

Γλώσσα Προγραμματισμού C

Γλώσσα Προγραμματισμού C

Η C είναι μια γλώσσα προγραμματισμού γενικής χρήσης, η οποία αναπτύχθηκε από τον Dennis Ritchie και τον Ken Thompson το 1972 στα εργαστήρια της AT & T Bell



Μεγάλο μέρος του UNIX γράφηκε στην C

Η C έγινε ιδιαίτερα δημοφιλής κυρίως για τον λιτό τρόπο σύνταξης και την ευκολία εκμάθησής της

Έχει δυνατότητες δομημένου προγραμματισμού και επιτρέπει τη χρήση υποπρογραμμάτων

Γλώσσα Προγραμματισμού C

Είναι γλώσσα υψηλού επιπέδου και μπορεί να λειτουργήσει σε διάφορες αρχιτεκτονικές υπολογιστών

Για να γίνεται αυτό είναι απαραίτητο να γίνεται μεταγλώττιση από τη γλώσσα προγραμματισμού στην εκάστοτε γλώσσα μηχανής

Η λειτουργία του αυτή εκτελείται από τον μεταγλωττιστή (compiler)

Ένα από τα χαρακτηριστικά της γλώσσας C είναι ότι μπορεί να χρησιμοποιηθεί και ως γλώσσα προγραμματισμού χαμηλού επιπέδου, επιτρέποντας άμεση πρόσβαση στους πόρους του υπολογιστή

Γλώσσα Προγραμματισμού C

Πλεονεκτήματα της C:

- ✓ Διαθέτει ευελιξία
- ✓ Έχει απλό συντακτικό
- ✓ Διαθέτει φορητότητα και ο κώδικας της μπορεί να μεταγλωττιστεί από διάφορους μεταγλωττιστές
- ✓ Είναι σχετικά μικρή και εύκολη στην εκμάθηση

Στάδια Ανάπτυξης Προγράμματος

Ένα πρόγραμμα αποτελείται από μια πεπερασμένη σειρά από εντολές

Αρχικά χρησιμοποιείται ένας συντάκτης κειμένου στον οποίο ο προγραμματιστής γράφει τον πηγαίο κώδικα (source code)



Το πρόγραμμα πρέπει να περάσει από την φάση της μεταγλώττισης η οποία εκτελείται από τον μεταγλωττιστή

Στάδια Ανάπτυξης Προγράμματος

Ο μεταγλωττιστής παράγει ένα αρχείο το οποίο ονομάζεται αρχείο αντικειμένου (object file) το οποίο περιέχει τον κώδικα σε γλώσσα μηχανής και έχει κατάληξη .obj

Κατά την φάση της μεταγλώττισης, ανιχνεύονται τα συντακτικά σφάλματα που έχει κάνει ο προγραμματιστής ο οποίος έχει παραβιάσει τους συντακτικούς κανόνες της γλώσσας

Τα λάθη αυτά πρέπει να διορθωθούν για να εκτελεστεί το πρόγραμμα

Στάδια Ανάπτυξης Προγράμματος

Στο επόμενο στάδιο, γίνεται η σύνδεση με την χρήση του συνδέτη ο οποίος αναζητά σε βιβλιοθήκες τις συναρτήσεις που χρησιμοποίησε ο προγραμματιστής

Ο συνδέτης παράγει ένα εκτελέσιμο αρχείο το οποίο έχει επέκταση .exe

Για να παραχθεί το αρχείο, θα πρέπει να ξεπεραστούν προβλήματα όπως για παράδειγμα η αδυναμία εντοπισμού μιας εξωτερικής μεταβλητής ή μιας συνάρτησης

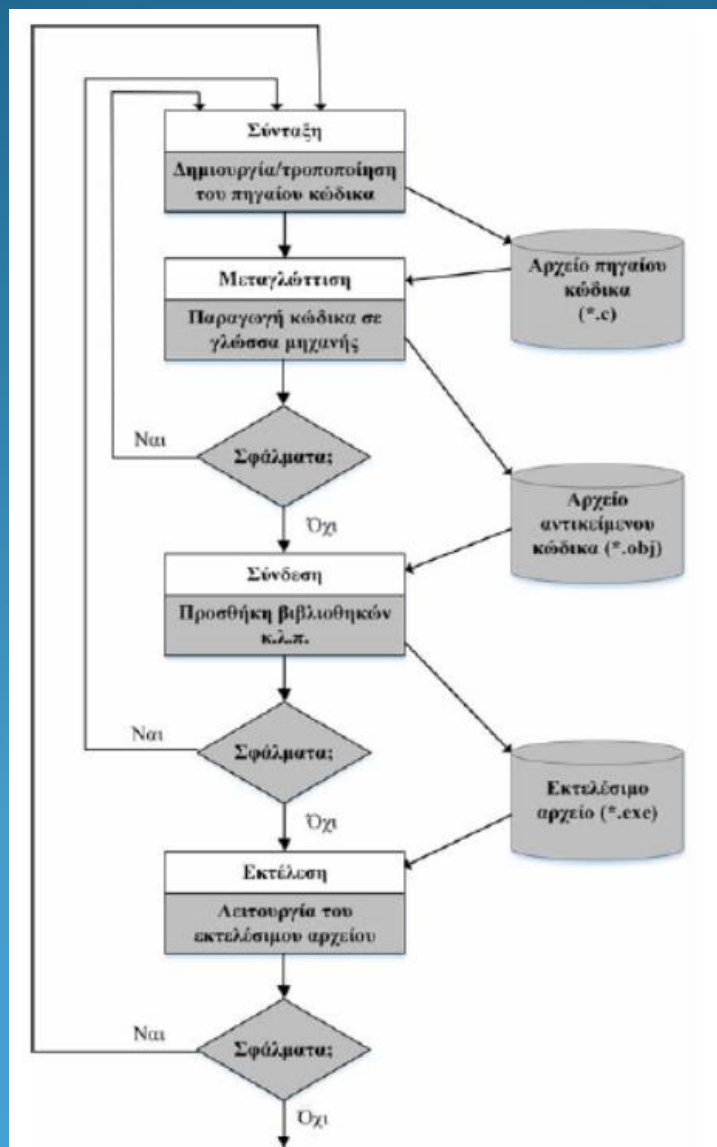
Στάδια Ανάπτυξης Προγράμματος

Στο τελικό στάδιο, εκτελείται το αρχείο .exe και γίνεται ο λογικός έλεγχος του προγράμματος

Ανιχνεύονται τα σημασιολογικά σφάλματα, με πιο χαρακτηριστικό παράδειγμα την διαίρεση ενός αριθμού με τον μηδέν, σφάλμα το οποίο δεν ανιχνεύεται κατά την μεταγλώττιση

Αν ο προγραμματιστής ανιχνεύσει κάποιο σφάλμα, τότε επιστρέφει στο πρώτο στάδιο και επαναλαμβάνει όλα τα στάδια

Στάδια Ανάπτυξης Προγράμματος



Πρόγραμμα στην C

Λέξεις - Κλειδιά

auto	double	int	struct	break	else	long
switch	case	enum	register	typedef	char	extern
return	union	const	float	short	unsigned	void
default	goto	sizeof	volatile	do	if	static
while						

Πρόγραμμα στην C

Το πρώτο πρόγραμμα

1. `//Hello World C πρόγραμμα`
2. `#include <stdio.h>`
3. `int main()`
4. `{`
5. `printf("Hello world\n");`
6. `return 0;`
7. `}`

Πρόγραμμα στην C

Σχόλια στην C

1. `// Hello World C πρόγραμμα`



Εισαγωγή σχολίων

Σχόλια μπορούν να γραφούν και σε περισσότερες από μια γραμμές και είναι το κείμενο ανάμεσα σε `/*` και `*/`

`/* Έτσι μπορούμε να γράφουμε σχόλια
και σε περισσότερες από μια γραμμές */`

Πρόγραμμα στην C

Οδηγίες προς τον προεπεξεργαστή

2. `#include <stdio.h>`



Ορίζουμε την βιβλιοθήκη χρησιμοποιώντας την λέξη `#include`

Σύνταξη: `#include <όνομα κεφαλίδας αρχείου.h>`

Μια βιβλιοθήκη είναι μια συλλογή χρήσιμων συναρτήσεων και συμβόλων

Η βιβλιοθήκη `stdio.h` είναι απαραίτητη για την εισαγωγή δεδομένων (`scanf()`) και εμφάνιση αποτελεσμάτων (`printf()`)

Πρόγραμμα στην C

Συνάρτηση main()

`int main()`



Κάθε πρόγραμμα έχει μια συνάρτηση με το όνομα `main()` όπου αρχίζει η εκτέλεση του προγράμματος

Σύνταξη: `int main() { εντολές }`

Πρόγραμμα στην C

Συνάρτηση printf()

```
printf ("Hello world\n");
```



Η συνάρτηση printf() εμφανίζει κάποιο μήνυμα ή το περιεχόμενο κάποιας μεταβλητής στην οθόνη του υπολογιστή

Για να εκτελεστεί η συνάρτηση printf() (όπως και η scanf()) είναι απαραίτητη η δήλωση της βιβλιοθήκης <stdio.h>

Μεταβλητές

Τα δεδομένα και τα αποτελέσματα ενός προγράμματος αποθηκεύονται στις μεταβλητές

Μία μεταβλητή μπορούμε να τη φανταστούμε σαν ένα κουτάκι, το οποίο έχει ένα όνομα και ένα περιεχόμενο κάθε στιγμή σε ένα πρόγραμμα, ΑΛΛΑ η τιμή αυτή μπορεί να αλλάζει κατά το τρέξιμο του προγράμματος

Η τιμή της μεταβλητής μπορεί να αλλάξει μέσα σε ένα αλγόριθμο ΑΛΛΑ δεν αλλάζει ποτέ ο τύπος της

Ονομασία Μεταβλητών

Για την ονοματοδοσία των μεταβλητών ακολουθούνται οι εξής κανόνες:

- ✓ Ένα όνομα μεταβλητής μπορεί να περιέχει τα γράμματα του αλφαβήτου, τα ψηφία 0-9 και τον χαρακτήρα υπογράμμισης _
- ✓ Το όνομα πρέπει να ξεκινά με γράμμα ή με χαρακτήρα υπογράμμισης
- ✓ Δεν μπορούμε να χρησιμοποιήσουμε δεσμευμένη λέξη

Ονομασία Μεταβλητών

Ποια είναι έγκυρα και μη έγκυρα ονόματα μεταβλητών:

- ✓ totalArea Έγκυρο
- ✓ \$product Μη Έγκυρο
- ✓ _temp_f Έγκυρο
- ✓ Counter1 Έγκυρο
- ✓ Total% Μη Έγκυρο
- ✓ 1counter Μη Έγκυρο

Σταθερές

Σταθερά είναι μία έκφραση που παραμένει αμετάβλητη σε όλη την διάρκεια της εκτέλεσης ενός αλγορίθμου

Παράδειγμα μιας σταθερής τιμής σε ένα πρόγραμμα μπορεί να είναι το ποσοστό ΦΠΑ

Τύποι Μεταβλητών

➤ Ακέραιες Μεταβλητές

Οι ακέραιες μεταβλητές δηλώνονται με την λέξη `int`

Για να εμφανίσουμε ή να διαβάσουμε κάποια ακέραια μεταβλητή, πρέπει να χρησιμοποιήσουμε τον προσδιοριστή `%d`

Με τους προσδιοριστές `%o`, `%x` μπορούμε να δούμε μια ακέραια μεταβλητή στην οκταδική και την δεκαεξαδική μορφή της

Τύποι Μεταβλητών

Υπάρχουν οι ακόλουθοι τύποι ακεραίων:

- ✓ Short (μέγεθος 2 bytes)
- ✓ Long (μέγεθος 4 bytes)
- ✓ Long long (μέγεθος 8 bytes)

Με την δήλωση οποιαδήποτε μεταβλητής παραχωρείται μνήμη για τη δημιουργία της μεταβλητής και ανάλογα με τον τύπο της προσδιορίζεται και ο απαιτούμενος χώρος μνήμης

Τύποι Μεταβλητών

➤ Μεταβλητές κινητής υποδιαστολής

Οι μεταβλητές κινητής υποδιαστολής δηλώνονται με την λέξη float

Εάν πρόκειται να αναγνωσθεί αριθμός διπλής ακριβείας, απαιτείται η χρήση του προσδιοριστή %f

Το μέγεθος των float αριθμών είναι 32 bits

Τύποι Μεταβλητών

➤ Μεταβλητές διπλής ακρίβειας

Οι μεταβλητές διπλής ακρίβειας δηλώνονται με την λέξη `double`

Στην ανάγνωση αριθμού διπλής ακριβείας, απαιτείται η χρήση του προσδιοριστή `%lf`

Στην ανάγνωση αριθμού εκτεταμένης ακριβείας, απαιτείται η χρήση του προσδιοριστή `%Lf`

Το μέγεθος των `double` αριθμών είναι 64 bits

Τύποι Μεταβλητών

➤ Μεταβλητές χαρακτήρα

Οι μεταβλητές χαρακτήρα δηλώνονται με την λέξη `char`

Εάν πρόκειται να αναγνωσθεί χαρακτήρας, απαιτείται η χρήση του προσδιοριστή `%c`

Αν χρησιμοποιηθεί ο προσδιοριστής `%d` με έναν χαρακτήρα, τότε θα τυπωθεί ο ASCII κωδικός του

Κώδικας ASCII

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	<u>NUL</u> 0000	<u>STX</u> 0001	<u>SOT</u> 0002	<u>ETX</u> 0003	<u>EOT</u> 0004	<u>ENQ</u> 0005	<u>ACK</u> 0006	<u>BEL</u> 0007	<u>BS</u> 0008	<u>HT</u> 0009	<u>LF</u> 000A	<u>VT</u> 000B	<u>FF</u> 000C	<u>CR</u> 000D	<u>SO</u> 000E	<u>SI</u> 000F
10	<u>DLE</u> 0010	<u>DC1</u> 0011	<u>DC2</u> 0012	<u>DC3</u> 0013	<u>DC4</u> 0014	<u>NAK</u> 0015	<u>SYN</u> 0016	<u>ETB</u> 0017	<u>CAN</u> 0018	<u>EM</u> 0019	<u>SUB</u> 001A	<u>ESC</u> 001B	<u>FS</u> 001C	<u>GS</u> 001D	<u>RS</u> 001E	<u>US</u> 001F
20	<u>SP</u> 0020	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	:	;	<	=	>	?
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	 007C	} 007D	~ 007E	<u>DEL</u> 007F
80																
90																
A0	<u>NBSP</u> 00A0	' 02BD	' 02BC	£ 00A3			¡ 00A6	§ 00A7	¨ 00A8	© 00A9		« 00AB	¬ 00AC	— 00AD		— 2015
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 0384	µ 0385	¶ 0386	· 00B7	¸ 0388	¹ 0389	ª 038A	» 00BB	¼ 038C	½ 00BD	¾ 038E	Ω 038F
C0	í 0390	À 0391	Á 0392	Â 0393	Ã 0394	Ä 0395	Å 0396	Æ 0397	Ç 0398	È 0399	É 039A	Ê 039B	Ë 039C	Ì 039D	Í 039E	Î 039F
D0	Π 03A0	Ρ 03A1		Σ 03A3	Τ 03A4	Υ 03A5	Φ 03A6	Χ 03A7	Ψ 03A8	Ω 03A9	Ϊ 03AA	Ϋ 03AB	ά 03AC	έ 03AD	ή 03AE	ί 03AF
E0	Ϊ 03B0	α 03B1	β 03B2	γ 03B3	δ 03B4	ε 03B5	ζ 03B6	η 03B7	θ 03B8	ι 03B9	κ 03BA	λ 03BB	μ 03BC	ν 03BD	ξ 03BE	ο 03BF
F0	π 03C0	ρ 03C1	ς 03C2	σ 03C3	τ 03C4	υ 03C5	φ 03C6	χ 03C7	ψ 03C8	ω 03C9	ϊ 03CA	ϋ 03CB	ό 03CC	ύ 03CD	ώ 03CE	

Συνάρτηση printf()

Η συνάρτηση printf() χρησιμοποιείται για την εμφάνιση μορφοποιούμενης πληροφορίας στην οθόνη

Για να χρησιμοποιηθεί η συνάρτηση printf() απαιτείται η ενσωμάτωση στο πρόγραμμα– μέσω της εντολής προεπεξεργαστή include– του αρχείου κεφαλίδας stdio.h

Συνάρτηση printf()

Οι μη εκτυπώσιμοι χαρακτήρες και οι αντίστοιχες ακολουθίες διαφυγής εμφανίζονται στον παρακάτω πίνακα:

Χαρακτήρας	Ακολουθία	Χαρακτήρας	Ακολουθία
Νέα γραμμή	\n	Μονό εισαγωγικό	\'
Οριζόντιος στηλοθέτης	\t	Διπλό εισαγωγικό	\"
Κατακόρυφος στηλοθέτης	\v	Πλάγια γραμμή	\\
Αλλαγή σελίδας	\f	Λατινικό ερωτηματικό	\?
Επαναφορά κεφαλής	\r	Οκταδικός αριθμός	\ooo
Οπισθοχώρηση μια θέση	\b	16δικός αριθμός	\xhhh
Συναγερμός	\a	Μηδενικός χαρακτήρας	\o

Συνάρτηση printf()

Ένα επιπλέον εργαλείο μορφοποίησης αποτελούν οι σημαίες (flags).

Σημαία	Λειτουργία
-	Αριστερή στοίχιση της εξόδου στο πεδίο πλάτους
+	Προσθήκη του προσήμου στις θετικές τιμές
#o	Προσθήκη του o μπροστά από οκταδικούς αριθμούς
#x	Προσθήκη του x μπροστά από δεκαεξαδικούς αριθμούς
#X	Προσθήκη του X μπροστά από δεκαεξαδικούς αριθμούς
o	Προσθήκη των απαιτούμενων μηδενικών μπροστά από την τιμή ώστε να καλυφθεί το πεδίο του πλάτους
Κενός χαρακτήρας	Προσθήκη του κενού χαρακτήρα μπροστά από τις μηδενικές τιμές

Άσκηση στην C

Να φτιάξετε πρόγραμμα στην c το οποίο:

- A) Να ορίζει μια μεταβλητή κινητής μεταβλητής στην τιμή 72.87
- B) Να εκτυπώνει ένα μήνυμα «Ο αριθμός είναι » και να εμφανίζει το ακέραιο μέρος με 8 ψηφία και το δεκαδικό με 4 ψηφία
- Γ) Να εκτυπώνει ένα μήνυμα «Ο αριθμός είναι » και να εμφανίζει το ακέραιο μέρος με 12 ψηφία και το δεκαδικό με 1 ψηφίο

Συνάρτηση scanf()

Η συνάρτηση scanf() χρησιμοποιείται για το διάβασμα δεδομένων από το πληκτρολόγιο

```
scanf("%d", &number)
```

Για να χρησιμοποιηθεί η συνάρτηση scanf() απαιτείται η ενσωμάτωση στο πρόγραμμα– μέσω της εντολής προεπεξεργαστή include– του αρχείου κεφαλίδας stdio.h

Συνάρτηση scanf()

Με την scanf() υπάρχει η δυνατότητα να χρησιμοποιήσουμε τον προσδιοριστή πλάτους

```
scanf("%4d", &number);
```

Αν ο χρήστης πληκτρολογήσει τον αριθμό 123456789, στην μεταβλητής number θα αποθηκευτούν τα 4 πρώτα ψηφία, δηλαδή η μεταβλητή number θα έχει περιεχόμενο την τιμή 1234

Συνάρτηση scanf()

Η scanf() είναι να θέσει περιορισμούς στους χαρακτήρες που μπορεί να διαβάσει

Αυτό γίνεται με την χρήση του προσδιοριστή []

Με τον προσδιοριστή %[A-D] επιτρέπει την καταχώρηση των χαρακτήρων 'Α', 'Β', 'Γ', 'Δ'

Με τον προσδιοριστή %[^A-D] δεν επιτρέπεται η ανάγνωση των χαρακτήρων αυτών

Άσκηση στην C

Να φτιάξετε πρόγραμμα στην c το οποίο:

- A) Να διαβάσει μια μεταβλητή χαρακτήρα
- B) Να εκτυπώνει ένα μήνυμα «Ο χαρακτήρας είναι: » και να εμφανίζεται ο χαρακτήρας που διαβάστηκε
- Γ) Να εκτυπώνει ένα νέο μήνυμα σε καινούργια σειρά το οποίο να λέει «Ο ASCII κωδικός του χαρακτήρα A (όποιος διαβάστηκε) είναι ο 65»

Άσκηση στην C

Να φτιάξετε πρόγραμμα στην c το οποίο:

- A) Να διαβάσει μια ακέραια μεταβλητή
- B) Να εκτυπώνει ένα μήνυμα «Η μεταβλητή που διαβάστηκε είναι: » και να εμφανίζεται ο αριθμός που διαβάστηκε
- Γ) Να εκτυπώνει ένα νέο μήνυμα σε καινούργια σειρά το οποίο να λέει «Ο αριθμός 46 έχει οκταδικό ισοδύναμο 56 και δεκαεξαδικό ισοδύναμο 2E»
- Δ) Να εμφανίζει σε διαφορετική σειρά το μέγεθος των τύπων μεταβλητής int, short, long και long long με την χρήση της συνάρτησης sizeof()

Άσκηση στην C

Να φτιαχτεί πρόγραμμα που να επιτελεί τα παρακάτω:

- A) Ζήτησε από τον χρήστη έναν χαρακτήρα
- B) Πάρε από τον χρήστη τον χαρακτήρα
- Γ) Τύπωσε τον χαρακτήρα και τον ASCII κωδικό του
- Δ) Βρες τον επόμενο χαρακτήρα
- E) Τύπωσέ τον μαζί με τον κωδικό του

Άσκηση στην C

Να φτιάξετε πρόγραμμα στην c το οποίο:

- A) Να ορίσετε σε μια ακέραια μεταβλητή την τιμή 323
- B) Να εκτυπώσετε σε διαφορετικές γραμμές το τι θα εμφανιστεί, χρησιμοποιώντας καθεμία από τις σημαίες που υπάρχουν στην c

Άσκηση στην C

Να φτιάξετε πρόγραμμα στην c το οποίο:

- A) Να διαβάσει και να εμφανίζει 2 ακέραιους αριθμούς
- B) Να διαβάσει και να εμφανίζει έναν χαρακτήρα
- Γ) Να διαβάσει και να εμφανίζει ένα ακέραιο αριθμό
- Δ) Να διαβάσει και να εμφανίζει 2 χαρακτήρες

Τελεστές

Οι τελεστές χρησιμοποιούνται για τη δημιουργία εκφράσεων

Μια έκφραση αποτελείται από έναν ή περισσότερους τελεστές και από έναν ή περισσότερους τελεστές

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με τον αριθμό των τελεστών στους οποίους επιδρούν:

- ✓ Μοναδιαίους (unary)
- ✓ Δυαδικούς (binary)
- ✓ Τριαδικούς (ternary)

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ **Αριθμητικοί τελεστές**

Τελεστής	Σύμβολο
Πολλαπλασιασμός	*
Διαίρεση	/
Υπόλοιπο διαίρεσης ακεραίων	%
Πρόσθεση	+
Αφαίρεση	-

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ **Αριθμητικοί τελεστές**

Να υπολογίσετε την τιμή των παρακάτω εκφράσεων:

☐ $X = 23 - 2 * 5$

☐ $X = 23 - (2 * 5)$

☐ $X = 23 \% 2 + 5$

☐ $X = 40 + 30 / 2$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ Σχεσιακοί τελεστές

Τελεστής	Σύμβολο
Ίσο	==
Διάφορο	!=
Μεγαλύτερο	>
Μεγαλύτερο ή ίσο	>=
Μικρότερο	<
Μικρότερο ή ίσο	<=

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ Σχεσιακοί τελεστές

Το αποτέλεσμα των σχεσιακών τελεστών είναι Αληθές (ακέραιο 1) ή Ψευδές (ακέραιο 0)

☐ $3 < 2$ Ψευδές

☐ $3 == 3$ Αληθές

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ **Λογικοί τελεστές**

Λογική Πράξη	Σύμβολο
Λογική σύζευξη (και)	&&
Λογική διάζευξη (ή)	
Λογική άρνηση (Όχι)	!

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ Λογικοί τελεστές

X	Y	X && Y	X Y	!X
Ψευδές	Ψευδές	Ψευδές	Ψευδές	Αληθές
Ψευδές	Αληθές	Ψευδές	Αληθές	Αληθές
Αληθές	Ψευδές	Ψευδές	Αληθές	Ψευδές
Αληθές	Αληθές	Αληθές	Αληθές	Ψευδές

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ Λογικοί τελεστές

Για $X=16$ και $Y=-6$ να υπολογιστούν οι εκφράσεις:

$$(X + 9) < (13 - Y) \quad \rightarrow \quad (16 + 9) < (13 - (-6))$$

$$\rightarrow 25 < 19 \quad \rightarrow \text{FALSE}$$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ Λογικοί τελεστές

Για $X=16$ και $Y=-6$ να υπολογιστούν οι εκφράσεις:

$(X < 5) \ || \ (Y-10) \rightarrow \text{FALSE}$

$-5 = X \ \acute{\eta} \ Y < X \rightarrow \text{FALSE}$

$(X \geq 7) \ \&\& \ (!((X-15) < Y) \rightarrow \text{TRUE}$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ Συνδυασμένοι τελεστές:

Πράξη	Σύμβολο	Παράδειγμα
$x = x + 2$	$+=$	$x += 2$
$x = x - 2$	$-=$	$x -= 2$
$x = x * 2$	$*=$	$x *= 2$
$x = x / 2$	$/=$	$x /= 2$
$x = x \% 2$	$\% =$	$x \% = 2$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ **Συνδυασμένοι τελεστές:**

$$X+=2 \quad \rightarrow X = X + 2$$

$$X*=50 \quad \rightarrow X = X * 50$$

$$X*=Y+5 \quad \rightarrow X = X * (Y + 5)$$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

- ✓ **Αυξητικοί και αφαιρετικοί τελεστές:**

Οι τελεστές αυτοί είναι μοναδιαίοι

Η λειτουργία τους είναι η αύξηση ή η μείωση του τελεστέου κατά μία μονάδα

Υπάρχουν δύο τρόποι χρήσης, είτε προθεματικά (prefix) είτε επιθεματικά (postfix) του τελεστέου

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ Αυξητικοί και αφαιρετικοί τελεστές:

Πράξη	Σύμβολο	Παράδειγμα (y=4)
X=X+1	++	x=y++; // x=4, y=5
		x=++y; //x=5,y=5
X=X-1	--	x=y--; // x=4, y=3
		x=--y; //x=3,y=3

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ **Αυξητικοί και αφαιρετικοί τελεστές:**

Να προσδιοριστεί η τιμή των X, Y και Z όταν οι τιμές των X και Y είναι 10 και 15 αντίστοιχα

$Z = ++X + Y$

$X = 11$

$Y = 15$

$Z = 26$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ **Αυξητικοί και αφαιρετικοί τελεστές:**

Να προσδιοριστεί η τιμή των X, Y και Z όταν οι τιμές των X και Y είναι 10 και 15 αντίστοιχα

$$Z = --X + Y$$

$$X = 9$$

$$Y = 15$$

$$Z = 24$$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ **Αυξητικοί και αφαιρετικοί τελεστές:**

Να προσδιοριστεί η τιμή των X, Y και Z όταν οι τιμές των X και Y είναι 10 και 15 αντίστοιχα

$$Z = X++ + Y$$

$$X = 11$$

$$Y = 15$$

$$Z = 25$$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ **Αυξητικοί και αφαιρετικοί τελεστές:**

Να προσδιοριστεί η τιμή των X, Y και Z όταν οι τιμές των X και Y είναι 10 και 15 αντίστοιχα

$$Z = X-- + Y$$

$$X = 9$$

$$Y = 15$$

$$Z = 25$$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ **Αυξητικοί και αφαιρετικοί τελεστές:**

Να προσδιοριστεί η τιμή των X, Y και Z όταν οι τιμές των X και Y είναι 10 και 15 αντίστοιχα

$$Z = X-- + ++Y$$

$$X = 9$$

$$Y = 16$$

$$Z = 26$$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ **Αυξητικοί και αφαιρετικοί τελεστές:**

Να προσδιοριστεί η τιμή των X, Y και Z όταν οι τιμές των X και Y είναι 10 και 15 αντίστοιχα

$Z = ++X + Y--$

$X = 11$

$Y = 14$

$Z = 26$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

✓ Τελεστές Διαχείρισης δυαδικών ψηφίων

Τελεστής	Δράση
&	Λογικό AND
	Λογικό OR
~	Συμπλήρωμα ως προς 1
^	Exclusive OR
<<	Ολίσθηση προς τα αριστερά
>>	Ολίσθηση προς τα δεξιά

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

- ✓ **Τελεστές Διαχείρισης δυαδικών ψηφίων**

Να υπολογιστούν οι εκφράσεις:

$10101010 \& 00001111 \rightarrow$

$(1 \& 0) (0 \& 0) (1 \& 0) (0 \& 0) (1 \& 1) (0 \& 1) (1 \& 1) (0 \& 1) \rightarrow$

00001010

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

- ✓ **Τελεστές Διαχείρισης δυαδικών ψηφίων**

Να υπολογιστούν οι εκφράσεις:

$10101010 \mid 00001111 \rightarrow 10101111$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

- ✓ **Τελεστές Διαχείρισης δυαδικών ψηφίων**

Να υπολογιστούν οι εκφράσεις:

$$10101010 \wedge 00001111 \rightarrow$$

$$(1 \wedge 0) (0 \wedge 0) (1 \wedge 0) (0 \wedge 0) (1 \wedge 1) (0 \wedge 1) (1 \wedge 1) (0 \wedge 1) \rightarrow$$

$$10100101$$

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

- ✓ **Τελεστές Διαχείρισης δυαδικών ψηφίων**

Να υπολογιστούν οι εκφράσεις:

$10 \ll 3 \rightarrow$

$00001010 \ll 3 \rightarrow 01010000$

Το δεκαδικό ισοδύναμο του 01010000 είναι το 80

Άρα το 10 πολλαπλασιάστηκε με το 8 (2^3)

Τελεστές

Οι τελεστές διακρίνονται ανάλογα με την λειτουργία τους διακρίνονται:

- ✓ **Τελεστές Διαχείρισης δυαδικών ψηφίων**

Να υπολογιστούν οι εκφράσεις:

$$40 \ll 2 \rightarrow$$

$$00101000 \ll 2 \rightarrow \quad 00001010$$

Το δεκαδικό ισοδύναμο του 00001010 είναι το 10

Άρα το 40 διαιρέθηκε με το 4 (2^2)

Μετατροπές Τύπων

Ένας τελεστής μπορεί να εκτελέσει λειτουργία μεταξύ 2 τελεστών διαφορετικού τύπου

- ✓ Έμμεση μετατροπή (μετατροπή από τον υπολογιστή)

Ο τύπος χαμηλότερης ιεραρχίας μετατρέπεται στον τύπο υψηλότερης ιεραρχίας χωρίς να υπάρχει απώλεια πληροφορίας

`char → int → long → float → double`

Σε μια πρόσθεση ενός `int` τελεστέου με έναν `float` τελεστή, ο `int` τελεστής μετατρέπεται σε `float` και εκτελείται η πρόσθεση

Μετατροπές Τύπων

- ✓ Γράψτε και τρέξτε το ακόλουθο πρόγραμμα

```
#include <stdio.h>
int main()
{
    char ch;
    int i;
    float fl;
    fl=i=ch='A';
    printf( "ch=%c, i=%d, fl=%2.2f,\n",ch,i,fl );
    ch=ch+1;
    i=fl+2*ch;
    fl=2.0*ch+i;
    printf( "ch=%c, i=%d, fl=%2.2f,\n",ch,i,fl );
    return 0;
}
```

Μετατροπές Τύπων

✓ Έμμεση μετατροπή (μετατροπή από τον υπολογιστή)

Στην εντολή `fl=i*ch='A'`; η μεταβλητή `i` παίρνει την τιμή 65 (ακέραια) ενώ η μεταβλητή `fl` παίρνει την τιμή 65.00 (κινητής υποδιαστολής)

Στην εντολή `i=fl+2*ch`; γίνεται η πράξη $65.00 + 2 * 66$

Το αποτέλεσμα μετατρέπεται σε `int` (197) και αποθηκεύεται στην `i`

Στην εντολή `fl=2.0*ch+i`; γίνεται η πράξη $2.0 * 66 + 197$

Όλες οι μεταβλητές `int` μετατρέπονται σε `float` και το αποτέλεσμα αποθηκεύεται στην μεταβλητή `fl`

Μετατροπές Τύπων

✓ Ρητή μετατροπή (μετατροπή από τον προγραμματιστή)

Ο προγραμματιστής χρησιμοποιεί τον τελεστή μετατροπής τύπου ή `cast` τελεστής

Έχει την μορφή (τύπος δεδομένων) π.χ. `(float)`

Ο `cast` τελεστής είναι μοναδιαίος

Μετατροπές Τύπων

✓ Ρητή μετατροπή (μετατροπή από τον προγραμματιστή)

```
int i,j;  
float f1,f2;  
i=5;  
j=2;  
f1=(float)i/(float)j+0.5;  
f2=i/j+0.5;
```

Στην εντολή `f1=(float)i/(float)j+0.5;`

Οι ακέραιες μεταβλητές `i` και `j` μετατρέπονται σε `float`, δηλαδή στις τιμές 5.0 και 2.0

Άρα στην διαίρεση $5.0 / 2.0$ το αποτέλεσμα που προκύπτει είναι 2.5

Άρα το αποτέλεσμα στην `f1` είναι $2.5 + 0.5 = 3.0$

Μετατροπές Τύπων

✓ Ρητή μετατροπή (μετατροπή από τον προγραμματιστή)

```
int i,j;  
float f1,f2;  
i=5;  
j=2;  
f1=(float)i/(float)j+0.5;  
f2=i/j+0.5;
```

Στην εντολή $f2=i / j+0.5$;

Οι ακέραιες μεταβλητές i και j έχουν τις τιμές 5 και 2

Άρα στην διαίρεση $5 / 2$ το αποτέλεσμα που προκύπτει είναι 2

Άρα το αποτέλεσμα στην $f2$ είναι $2 + 0.5 = 2.5$

Εκφράσεις

Μια έκφραση είναι ένας συνδυασμός τελεστών και τελεστέων και έχει πάντα μια τιμή

Για να γίνει ο υπολογισμός μιας έκφρασης, οι τελεστές ταξινομούνται σε επίπεδα προτεραιότητας

Η ύπαρξη περισσότερων τελεστών στο ίδιο επίπεδο προτεραιότητας επιβάλλει τον προσδιορισμό της κατεύθυνσης εφαρμογής

Ένας τελεστής καλείται αριστερής προσηταιριστικότητας, όταν σε εκφράσεις που περιέχουν πολλά στιγμιότυπα του τελεστή η ομαδοποίηση γίνεται από τα αριστερά προς τα δεξιά

Το αποτέλεσμα είναι ο υπολογισμός $8-4-2$ να υπολογίζεται ως $(8-4)-2$

Εκφράσεις

Τελεστές (με φθίνουσα σειρά προτεραιότητας)	Είδος προσηταιριστικότητα
() [] ->	Από αριστερά προς τα δεξιά
! ~ ++ -- &	Από δεξιά προς τα αριστερά
sizeof()	
*a / %	
+ -	
<< >>	
< <= > >=	
== ! =	Από αριστερά προς τα δεξιά
&	
^	
&&	
?:	
= += -= *= %= &= ^=	Από δεξιά προς τα αριστερά
=	
<<= >>=	Από αριστερά προς τα δεξιά

Δομή ελέγχου if

Στα περισσότερα προβλήματα, δεν ακολουθούνται τα ίδια βήματα κατά την επίλυση τους. Συνήθως, λαμβάνονται διαφορετικές αποφάσεις, ανάλογα με τις τιμές μιας ή περισσότερων μεταβλητών

Στη δομή επιλογής αρχικά ελέγχεται κάποια λογική συνθήκη. Ανάλογα με την τιμή της λογικής συνθήκης (αληθής ή ψευδής) ακολουθεί η εκτέλεση ή όχι μιας σειράς εντολών

Η δομή της απλής επιλογής είναι:

```
if (συνθήκη)
{
    ομάδα εντολών
}
```

Δομή ελέγχου if

Η δομή της απλής επιλογής είναι:

```
if (x < 0)
{
    x += 1;
}
```

Όταν πρόκειται να εκτελεστεί μια εντολή, τότε τα {} μπορούν να παραλειφθούν

```
if (x < 0)
    x += 1;
```

Δομή ελέγχου if

Η δομή της απλής if είναι:

```
if (x<0)
{
    scanf("%d", &x);
    printf("x= %d \n",x);
    x +=1;
}
```

Δομή ελέγχου if

Η δομή της if στην γενική της μορφή είναι:

```
if (συνθήκη)
{
    Ομάδα εντολών 1;
}
else
{
    Ομάδα εντολών 2;
}
```

Δομή ελέγχου if

Στην ομάδα εντολών 2 μπορεί να υπάρχει έλεγχος κάποιας άλλης συνθήκης:

```
if (συνθήκη)
{
    Ομάδα εντολών 1;
}
else if (συνθήκη 2)
{
    Ομάδα εντολών 3;
}
else if (συνθήκη 3)
{
    Ομάδα εντολών 4;
}
else
{
    Ομάδα εντολών 5;
}
```

Δομή ελέγχου if

Άσκηση 1: Να γράψεις πρόγραμμα το οποίο θα δέχεται ως είσοδο έναν αριθμό και θα μήνυμα αν είναι θετικός ή αρνητικός ή ίσος με το 0.

Άσκηση 2: Να γραφεί πρόγραμμα το οποίο θα δέχεται σαν είσοδο τα τέρματα που σημείωσε η γηπεδούχος και η φιλοξενούμενη ομάδα σε έναν αγώνα και να τυπώνει ανάλογα «1», «X» ή «2» (Νίκη γηπεδούχου, ισοπαλία, Νίκη φιλοξενούμενης αντίστοιχα) και να εμφανίζεται το σκορ με στυλ π.χ. 3-1

Άσκηση 3: Να γράψεις πρόγραμμα το οποίο θα δέχεται ως είσοδο έναν αριθμό και θα εμφανίζει την απόλυτη τιμή του.

Δομή ελέγχου if

Άσκηση 4: Ένα ηλεκτρονικό θερμόμετρο εμφανίζει τα ακόλουθα μηνύματα σε μια μικρή οθόνη:

A) «ΚΡΥΟ», όταν η θερμοκρασία είναι πιο χαμηλή από 10ο C

B) «ΜΕΤΡΙΑ», όταν η θερμοκρασία είναι υψηλότερη από 10ο C και 20ο C

Γ) «ΖΕΣΤΗ», όταν η θερμοκρασία είναι υψηλότερη από 20ο C

Να γραφεί πρόγραμμα που θα διαβάζει την θερμοκρασία και θα εμφανίζει τα αντίστοιχα μηνύματα.

Δομή ελέγχου if

Άσκηση 5:

Σε μια εταιρία, οι υπάλληλοι παίρνουν μηνιαίο οικογενειακό επίδομα ανάλογα με τον αριθμό των παιδιών που έχουν, όπως παρακάτω:

- ✓ Για 1 παιδί , επίδομα 30 €
- ✓ Για 2 παιδιά , επίδομα 60 €
- ✓ Για 3 παιδιά , επίδομα 120 €

Στον τελικό μισθό γίνονται 20% κρατήσεις.

Να γίνει πρόγραμμα που θα:

- A) διαβάζει το βασικό μισθό και τον αριθμό των παιδιών ενός υπαλλήλου,
- B) υπολογίζει το επίδομα που θα καταβληθεί και
- Γ) εμφανίζει τις μηνιαίες αποδοχές το υπαλλήλου

Υποθετικός τελεστής

Ο υποθετικός τελεστής αποτελείται από 2 σύμβολα (? :)

Είναι ο μοναδικός τριαδικός τελεστής στην C

Η σύνταξη του είναι η ακόλουθη:

Έκφραση 1 ? έκφραση 2 : έκφραση 3

Ελέγχεται η έκφραση 1.

- ✓ Αν είναι αληθής, εκτελείται η έκφραση 2
- ✓ Αν είναι ψευδής, εκτελείται η έκφραση 3

Υποθετικός τελεστής

Ο υποθετικός τελεστής αποτελείται από 2 σύμβολα (? :)

Είναι ο μοναδικός τριαδικός τελεστής στην C

Η σύνταξη του είναι η ακόλουθη:

Έκφραση 1 ? έκφραση 2 : έκφραση 3

Ελέγχεται η έκφραση 1.

- ✓ Αν είναι αληθής, εκτελείται η έκφραση 2
- ✓ Αν είναι ψευδής, εκτελείται η έκφραση 3

Υποθετικός τελεστής

$Z = X > Y ? 10 : 20$

Ελέγχεται αν το X είναι μεγαλύτερο του Y

Αν το X είναι μεγαλύτερο του Y, τότε το Z παίρνει την τιμή 10

Αν το X είναι μικρότερο ή ίσο του Y, τότε το Z παίρνει την τιμή 20

Υποθετικός τελεστής

Άσκηση:

Να γραφεί πρόγραμμα το οποίο:

- ✓ Θα εμφανίζει μήνυμα και θα διαβάσει 2 αριθμούς
- ✓ Θα εμφανίζει μήνυμα για το ποιος από τους 2 αριθμούς είναι μεγαλύτερος.

Σημ1: Το πρόγραμμα θα γίνει και με την δομή if και με τον Υποθετικό Τελεστή

Σημ2: Θεωρείστε ότι δεν θα δοθούν ίσες τιμές

Υποθετικός τελεστής

Άσκηση:

Να γράψεις πρόγραμμα το οποίο θα δέχεται ως είσοδο έναν αριθμό και θα εμφανίζει την απόλυτη τιμή του χρησιμοποιώντας τον Υποθετικό τελεστή.

Δομή switch

Η switch μας επιτρέπει την διακλάδωση του προγράμματος σε διαφορετικά block κώδικα ανάλογα με την περίπτωση που ισχύει

Επειδή όμως οι περιπτώσεις της switch είναι σταθερές και όχι λογικές εκφράσεις, μπορεί να ισχύει μόνο μια περίπτωση, και άρα να εκτελεστεί μόνο ένα τμήμα κώδικα

Σύνταξη:

```
switch (μεταβλητή) {  
    case σταθερά1: εντολή_A;  
        break;  
    case σταθερά2: εντολή_B;  
        break;  
    case σταθερά3: εντολή_Γ;  
        break;  
    default:      εντολή_Δ;  
}
```

Δομή switch

Παράδειγμα

```
switch (x) {  
    case 1: x++;  
        break;  
    case 2: x--;  
        break;  
    case 3: x+=x;  
        break;  
    default: printf("%d", x);  
}
```

Ανάλογα την τιμή του x θα εκτελεστεί και το κατάλληλο τμήμα εντολών:

- ✓ Αν το x είναι 1, τότε θα εκτελεστεί η x++
- ✓ Αν το x είναι 2, τότε θα εκτελεστεί η x--
- ✓ Αν το x είναι 3, τότε θα εκτελεστεί η x+=x
- ✓ Σε ΟΛΕΣ τις άλλες περιπτώσεις θα εκτελεστεί η printf("%d",x);

Δομή switch

Άσκηση:

Να γραφεί πρόγραμμα με την δομή switch το οποίο:

- ✓ Θα εμφανίζει μήνυμα και θα διαβάζει 1 αριθμό
- ✓ Θα εμφανίζει μήνυμα:
 - ❖ Αν είναι 0, να εμφανίζεται Χειμώνας
 - ❖ Αν είναι 1, να εμφανίζεται Άνοιξη
 - ❖ Αν είναι 2, να εμφανίζεται Καλοκαίρι
 - ❖ Αν είναι 3, να εμφανίζεται Φθινόπωρο
 - ❖ Σε κάθε άλλη περίπτωση να εμφανίζεται ότι δόθηκε λάθος νούμερο

Σημ1: Το πρόγραμμα θα γίνει και με την δομή if και με τον Υποθετικό Τελεστή

Δομή switch

Άσκηση:

Να γραφεί πρόγραμμα με την δομή switch το οποίο:

- ✓ Θα εμφανίζει μήνυμα και θα διαβάζει 1 χαρακτήρα που θα είναι ένα από τα τέσσερα σύμβολα της αριθμητικής (+ , - , * , /)
- ✓ Θα εμφανίζει μήνυμα και θα διαβάζει 2 ακέραιους αριθμούς
- ✓ Ανάλογα με το σύμβολο θα εκτελεί και την αντίστοιχη πράξη
- ✓ Σε περίπτωση που δοθεί ως χαρακτήρας κάποιο άλλος χαρακτήρας εκτός από τα 4 σύμβολα της αριθμητικής, θα εμφανιστεί μήνυμα ότι δόθηκε λάθος σύμβολο

Σημ1: Στην πράξη της διαίρεσης, θα πρέπει να λάβετε υπόψη σας ότι η διαίρεση μπορεί να εκτελεστεί ΜΟΝΟ αν ο παρανομαστής είναι διαφορετικός του 0 (μηδέν)

Σημ2: Η μεταβλητή του αποτελέσματος θα φροντίσετε να είναι τύπου float

Σημ3: Στην διαίρεση θέλουμε να δούμε δεκαδικό αποτέλεσμα π.χ. $9/2=4.5$

ΑΣΚΗΣΗ

Ένα κατάστημα κοστολογεί τους υπολογιστές του, ανάλογα με τον αριθμό Η/Υ της παραγγελίας. Η κοστολόγηση των Η/Υ γίνεται με βάση τον παρακάτω πίνακα:

Ποσότητα	Τιμή κάθε υπολογιστή
1-30	600€
31-70	500€
71-100	450€
101-200	440€
Πάνω από 200	400€

Στην τελική τιμή των Η/Υ προστίθεται και ο ΦΠΑ (23%). Να γράψεις αλγόριθμο ο οποίος θα διαβάζει τον αριθμό των Η/Υ της παραγγελίας και θα εμφανίζει το κόστος των Η/Υ. Επιπλέον, θα υπολογίζει την αξία του ΦΠΑ και την τελική τιμή των Η/Υ. Η εμφάνιση της εξόδου θα έχει την παρακάτω μορφή:

Το κόστος των 170 Η/Υ είναι x€.

Η αξία του ΦΠΑ είναι y€.

Η τελική αξία της παραγγελίας είναι w€.

Δομή switch

Παράδειγμα

```
switch (x) {  
    case 1: x++;  
        break;  
    case 2: x--;  
        break;  
    case 3: x+=x;  
        break;  
    default: printf("%d", x);  
}
```

Ανάλογα την τιμή του x θα εκτελεστεί και το κατάλληλο τμήμα εντολών:

- ✓ Αν το x είναι 1, τότε θα εκτελεστεί η x++
- ✓ Αν το x είναι 2, τότε θα εκτελεστεί η x--
- ✓ Αν το x είναι 3, τότε θα εκτελεστεί η x+=x
- ✓ Σε ΟΛΕΣ τις άλλες περιπτώσεις θα εκτελεστεί η printf("%d",x);

Δομή While

Η δομή επανάληψης While είναι βρόχος με συνθήκη εισόδου, οδηγούμενη από κάποιο γεγονός

```
While (συνθήκη)
{
    Ομάδα εντολών
}
```

Λειτουργία δομής:

- ✓ Έλεγε τη συνθήκη
- ✓ Αν είναι αληθής, τότε εκτέλεσε την ομάδα εντολών
- ✓ Αν είναι ψευδής, τότε σταμάτησε τη λειτουργία του βρόχου

Δομή While

Η δομή επανάληψης While είναι κατάλληλη στις περιπτώσεις που δεν είναι γνωστός ο αριθμός των επαναλήψεων

Όταν η ομάδα εντολών αποτελείται από 1 εντολή δεν απαιτούνται τα { }

Δομή While

Γράψτε το ακόλουθο πρόγραμμα:

```
#include <stdio.h>

int main()
{
    int counter=10;
    int counterLimit=20;
    while (counter<counterLimit)
    {
        counter++;
        printf( "counter is %d\n",counter );
    }

    return 0;
}
```

Δομή While

Γράψε ένα πρόγραμμα που υπολογίζει το άθροισμα των αριθμών από το 1 μέχρι το 100, αλλά προσθέτει μόνο τους αριθμούς που είναι ζυγοί.

Δομή While

Γράψε ένα πρόγραμμα που ζητά από τον χρήστη να εισάγει έναν αριθμό και συνεχίζει να ζητάει αριθμούς μέχρι ο χρήστης να εισάγει έναν αριθμό που είναι διαφορετικός από το μηδέν. Το πρόγραμμα θα εμφανίζει πόσοι θετικοί και πόσοι αρνητικοί αριθμοί δόθηκαν.

Δομή While

Γράψε ένα πρόγραμμα που θα ζητάει από τον χρήστη να εισάγει αριθμούς μέχρι να εισάγει το 0. Το πρόγραμμα θα βρει τον μεγαλύτερο και τον μικρότερο αριθμό από αυτούς που έχουν εισαχθεί και θα τους εκτυπώσει

Δομή While

Να γραφεί πρόγραμμα σε C που να υπολογίζει τη μέση τιμή ακεραίων αριθμών, όταν εισάγονται από το πληκτρολόγιο. Η εισαγωγή τερματίζεται ,όταν δοθεί σαν τιμή ακέραιου το 0.

(Θέμα Πιστοποίησης)

Δομή Do-While

Η δομή επανάληψης Do - While μοιάζει με την `while`, αλλά με μία πολύ σημαντική διαφορά, καθώς το μπλοκ εντολών εκτελείται τουλάχιστον μία φορά, γιατί η συνθήκη ελέγχεται στο τέλος του βρόχου

```
do  
{  
    Ομάδα εντολών  
} while (συνθήκη);
```

Λειτουργία δομής:

- ✓ Εκτελείται η ομάδα εντολών 1 φορά
- ✓ Ελέγχεται η συνθήκη στο while
- ✓ Αν η συνθήκη είναι αληθής, ο βρόχος επαναλαμβάνεται
- ✓ Αν η συνθήκη είναι ψευδής, ο βρόχος τερματίζεται

Δομή Do - While

Η δομή do-while προτιμάται, όταν θέλουμε να εκτελεστεί ο κώδικας τουλάχιστον μία φορά

Χρησιμοποιείται όταν πρόκειται να κάνουμε έλεγχο εγκυρότητας σε δεδομένα εισόδου

Είναι σημαντικό να θυμόμαστε το (;) μετά τη while(συνθήκη)

Δομή Do - While

Γράψτε το ακόλουθο πρόγραμμα:

```
#include <stdio.h>

int main() {
    int i = 1;

    do {
        printf("%d\n", i);
        i++;
    } while (i <= 5);

    return 0;
}
```

Δομή Do-While

Γράψτε το ακόλουθο πρόγραμμα, το οποίο ζητάει έναν θετικό αριθμό:

```
#include <stdio.h>

int main() {
    int n;

    do {
        printf("Δώσε έναν θετικό αριθμό: ");
        scanf("%d", &n);
    } while (n <= 0);
}
```

Δομή While - Do While

Υποθέτοντας ότι μια πόλη έχει πληθυσμό 550.000 κατοίκους, με ετήσιο ρυθμό αύξησης 3,3%, να γραφεί πρόγραμμα C που θα υπολογίζει το πληθυσμό για τα επόμενα πέντε έτη.

(Θέμα Πιστοποίησης)

Δομή Do - While

Σε ένα πολυκατάστημα αποφασίστηκε να γίνεται έκπτωση στους πελάτες ανάλογα με το ποσό των αγορών τους, με βάση τον παρακάτω πίνακα:

Ποσό αγορών	Έκπτωση
έως και 300 €	2%
πάνω από 300 έως και 400 €	5%
πάνω από 400 €	7%

Να γραφεί πρόγραμμα που:

α. για κάθε πελάτη,

1. να διαβάζει το αρχικό του επιθέτου του και το ποσό των αγορών του.
2. να υπολογίζει την έκπτωση που δικαιούται.
3. να εμφανίζει το αρχικό του επιθέτου του και το ποσό που θα πληρώσει μετά την έκπτωση.

β. να επαναλαμβάνει τη διαδικασία μέχρι να δοθεί ως αρχικό του επιθέτου του ο χαρακτήρας q

γ. να εμφανίζει μετά το τέλος της διαδικασίας τη συνολική έκπτωση που έγινε για όλους τους πελάτες

Δομή Do - While

Για κάθε μαθητή δίνονται τα στοιχεία: προφορικός και γραπτός βαθμός ενός μαθήματος.

Να γραφεί πρόγραμμα το οποίο εκτελεί τις ακόλουθες λειτουργίες:

α. Διαβάζει τα στοιχεία 5 μαθητών ελέγχοντας αν ο προφορικός και ο γραπτός βαθμός είναι από 0 μέχρι και 20 για κάθε μαθητή

β. Υπολογίζει τον τελικό βαθμό του μαθήματος, ο οποίος είναι το άθροισμα του 30% του προφορικού βαθμού και του 70% του γραπτού βαθμού. Επίσης, τυπώνει το ονοματεπώνυμο του μαθητή και τον τελικό βαθμό του μαθήματος.

γ. Υπολογίζει και τυπώνει το πλήθος των μαθητών που έχουν βαθμό μεγαλύτερο του 18

Διαφορά break - continue

```
#include <stdio.h>
```

```
int main() {  
    int i=1;  
    printf("Px break:\n");  
    while (i <=10)  
    {  
        if (i == 5) {  
            break;  
        }  
        printf("%d ", i);  
        i++;  
    }  
    printf("\n\nPx continue:\n");  
    i=1;  
    while ( i<=10)  
        { if (i == 5) {  
            i++;  
            continue;  
        }  
        printf("%d ", i);  
        i++;  
    }  
    return 0;  
}
```

Διαφορά break - continue

Break

- ✓ Τερματίζει εντελώς μια δομή επανάληψης (for, while, do-while).
- ✓ Η εκτέλεση συνεχίζεται μετά το loop.

Continue

- ✓ Δεν τερματίζει την επανάληψη.
- ✓ Παραλείπει τον τρέχοντα κύκλο και προχωρά στον επόμενο..

Δομή While

Να γραφεί πρόγραμμα, το οποίο θα επιτελεί τα ακόλουθα:

- (α) Θα διαβάζει διαδοχικά χαρακτήρες, έως ότου πληκτρολογηθεί ο χαρακτήρας q.
- (β) Θα μετρά το κενά.
- (γ) Μετά το πέρας της ανάγνωσης θα εμφανίζει στην οθόνη τον αριθμό τους.