ΟΝΟΜΑΤΕΠΩΝΥΜΟ: .....
SDI (sdiYYOONNN): .....
GitHub username: .....

Εισαγωγή στον Προγραμματισμό - Ιανουάριος 2026

Διάρκεια: 135 λεπτά / Σύνολο: 110 Μονάδες

Τα προγράμματα C που θα γράψετε πρέπει να είναι δομημένα, διατυπωμένα ευκρινώς εντός του διαθέσιμου χώρου και με επαρκή τεκμηρίωση ώστε να είναι κατανοητά.

1. Mystery [10 Μονάδες]

Η συνάρτηση `mystery` δέχεται ως όρισμα έναν ακέραιο αριθμό που αποτελείται από τα 3 τελευταία ψηφία του sdi σας. Για παράδειγμα, αν ο sdi σας είναι ο sdi2500789 τότε καλούμε την συνάρτηση ως `mystery(789)`. Τι θα τυπώσει η συνάρτηση για τα ψηφία του δικού σας sdi; Αν δεν έχετε sdi, επιλέξτε έναν τυχαίο τριψήφιο. Αιτιολογήστε όπου νομίζετε πως χρειάζεται.

```
int mystery(int number) {
  int seed = number % 10;
  int count = (seed == 0 || seed > 4) ? 2 : seed;
  printf("Seed %d, count: %d\n", seed, count);
  for(int i = 0; i < count; ++i) {
    printf("Loop %d: %d\n", i, ++seed);
    seed &= 0xF;
  }
  printf("Result: %d\n", seed);
  return seed;
}
```

Απάντηση:

---

2. Η συνάρτηση compute (15 Μονάδες)

```
int *compute(const int *a, size_t na, const int *b, size_t nb) {
    size_t i = 0, j = 0, k = 0;
    int * out = malloc((na + nb) * sizeof(int));
    while (i < na && j < nb) {
        if (a[i] <= b[j])
            out[k++] = a[i++];
        else
            out[k++] = b[j++];
    }
    while (i < na)
        out[k++] = a[i++];
    while (j < nb)
        out[k++] = b[j++];
    return out;
}
```

Τι κάνει η συνάρτηση compute (μέχρι 15 λέξεις εξήγηση);

Ποιο θα είναι το περιεχόμενο της μεταβλητής result μετά την εκτέλεση της παρακάτω ακολουθίας εντολών και σε ποια κατηγορία μνήμης είναι αποθηκευμένο;

```
int arg1[] = {71, 111, 111, 100, 32, 0};
int arg2[5] = {'j', 111, 98, '!', 0};
int * result = compute(arg1, 3, arg2, 2);
```

Υπάρχει κάποιο σφάλμα στην συνάρτηση compute; Αιτιολογήστε την απάντησή σας.

3. Εύρεση Πρώτων Παραγόντων - factor [25 Μονάδες]

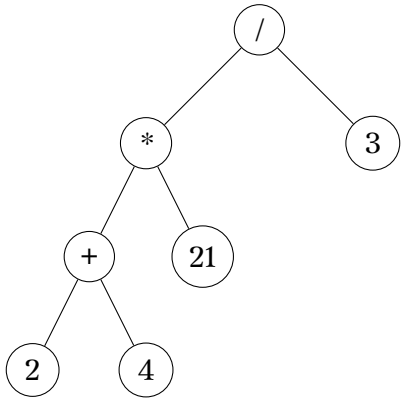
Κάθε φυσικός αριθμός μπορεί να γραφεί ως το γινόμενο των πρώτων παραγόντων του. Για παράδειγμα, ο αριθμός 42 μπορεί να γραφεί ως $2 \cdot 3 \cdot 7$. Γράψτε ένα πρόγραμμα το οποίο παίρνει έναν θετικό ακέραιο ως το πρώτο όρισμα στην γραμμή εντολών και τυπώνει στην έξοδο όλους τους πρώτους παράγοντές του. Παράδειγματα εκτέλεσης ακολουθούν:

```
$ ./factor 42
Factors of 42: 2 3 7
$ ./factor 54
Factors of 54: 2 3 3 3
```

Απάντηση:

4. Διερμηνέας Αριθμητικών Εκφράσεων - eval [25 Μονάδες]

Γράψτε μια συνάρτηση `eval` η οποία παίρνει ως όρισμα ένα δέντρο ακεραίων εκφράσεων τύπου `Expr` και επιστρέφει το αποτέλεσμα της αποτίμησης της έκφρασης. Για παράδειγμα, για την έκφραση (2 + 4) * 21 / 3 το δέντρο έκφρασης δείχνει ως εξής:



και αν δοθεί στην συνάρτηση eval, περιμένουμε να μας επιστραφεί η τιμή: $42 = (2+4)*21/3$.

Ποια είναι η χρονική και η χωρική πολυπλοκότητα του αλγορίθμου σας; (8/25)

Ποιον αλγόριθμο διάσχισης επιλέξατε και γιατί; (2/25)

Ο ορισμός του τύπου `Expr` δίνεται παρακάτω:

```
typedef enum {
  VALUE,  // current node is an integer value
  ADD,    // current node is addition of two nodes
  SUB,    // current node is subtraction of left minus right
  MUL,    // current node is the multiplication of two nodes
  DIV     // current node is the division of left by right
} exp_type;

typedef struct node {
  exp_type type;
  int value;
  struct node * left;
  struct node * right;
} * Expr;
```

Απάντηση:

5. Περικύκλωση - encirclement [25 Μονάδες]

Σε πολλά παιχνίδια στρατηγικής η περικύκλωση του αντιπάλου αφαιρεί τις δυνατότητες κίνησής του και πολύ συχνά οδηγεί σε γρήγορη νίκη. Πότε είναι όμως περικυκλωμένο ένα πιόνι του αντιπάλου; Το Σχήμα 1 δείχνει ένα παράδειγμα:

	A	B	C	D	E	F	G	H	I	J
10		O	O	O						
9	O				O					
8	O		X		O					
7	O					O	O	O		
6		O	O	O			X		O	
5					O		X		O	
4			O		X	O	O	O	O	
3		O	X	O						
2			O			O				
1					O	X	O			

Σχήμα 1: Τα πιόνια του μαύρου (X) στις θέσεις C3, C8, G5, G6 είναι περικυκλωμένα, ενώ στις θέσεις F1, E4 είναι ελεύθερα καθώς έχουν διέξοδο προς τα άκρα του πλέγματος.

Θεωρούμε πως ο χάρτης του παιχνιδιού είναι ένα τετραγωνικό πλέγμα και πως κάθε κελί: (1) είτε περιέχει μαύρα πιόνια (X), (2) είτε περιέχει λευκά πιόνια (O), (3) είτε είναι κενό (.). Ένα πιόνι θεωρείται περικυκλωμένο όταν δεν υπάρχει σειρά κινήσεων πάνω, κάτω, αριστερά, δεξιά (δεν επιτρέπονται διαγώνιες κινήσεις) που να επιτρέπει στο πιόνι να φτάσει στο άκρο του πλέγματος κινούμενο μόνο σε άδεια κελιά ή κελιά που περιέχουν το ίδιο χρώμα.

Γράψτε ένα πρόγραμμα το οποίο διαβάζει από την πρότυπη είσοδο (stdin) την διάσταση του πλέγματος και τα περιεχόμενα του χάρτη και τυπώνει στην πρότυπη έξοδο (stdout) για το κάθε μαύρο πιόνι αν είναι ελεύθερο ή όχι. Το πρόγραμμά σας πρέπει να είναι όσο πιο αποδοτικό γίνεται και να μπορεί να διαχειριστεί περιπτώσεις σφάλματος.

Ποια είναι η χρονική και η χωρική πολυπλοκότητα του αλγορίθμου σας; (8/25)

Παράδειγμα εκτέλεσης ακολουθεί:

```
$ cat map.txt
10
. . . . O X O . . .
. . O . . O . . . .
. O X O . . . . . .
. . O . X O O O O .
. . . . O . X . O .
. O O O . . X . O .
O . . . . O O O . .
O . X . O . . . . .
O . . . O . . . . .
. O O O . . . . . .
$ ./encirclement < map.txt
The following pawns are free: F1, E4
The following pawns are encircled: C3, G5, G6, C8
```

Απάντηση:

6. Ο Στέργιος Ξαναχτυπά [10 Μονάδες]

Ο Στέργιος κουράστηκε με τους μαθηματικούς γρίφους και αποφάσισε να ασχοληθεί με κώδικα συστημάτων. Ο μπαμπάς του—ο Μάκης—του έδωσε το ακόλουθο κομμάτι κώδικα:

```
#include <stdio.h>

int main(void) {
    char *p1 = NULL;
    char *p2 = NULL;
    int n = 16;
    while (n-- > 0) {
        if (*p1++ != *p2++) {
            printf("Not Equal\n");
        }
    }
    printf("Equal\n");
    return 0;
}
```

Τι θα κάνει το παραπάνω κομμάτι κώδικα; Θα τυπώσει Equal, Not Equal, δεν κάνει compile, θα έχει κάποιο runtime error ή δεν μπορούμε να γνωρίζουμε; Αιτιολογήστε την απάντησή σας. Μη σωστά αιτιολογημένες απαντήσεις δεν θα λάβουν βαθμολογία.

Απάντηση:

Βοηθήματα

ASCII Table

Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec
NUL	0	NAK	21	*	42	?	63	T	84	i	105
SOH	1	SYN	22	+	43	@	64	U	85	j	106
STX	2	ETB	23	,	44	A	65	V	86	k	107
ETX	3	CAN	24	-	45	B	66	W	87	l	108
EOT	4	EM	25	.	46	C	67	X	88	m	109
ENQ	5	SUB	26	/	47	D	68	Y	89	n	110
ACK	6	ESC	27	0	48	E	69	Z	90	o	111
BEL	7	FS	28	1	49	F	70	[	91	p	112
BS	8	GS	29	2	50	G	71	\	92	q	113
HT	9	RS	30	3	51	H	72	]	93	r	114
LF	10	US	31	4	52	I	73	^	94	s	115
VT	11	Space	32	5	53	J	74	_	95	t	116
FF	12	!	33	6	54	K	75	`	96	u	117
CR	13	"	34	7	55	L	76	a	97	v	118
SO	14	#	35	8	56	M	77	b	98	w	119
SI	15	$	36	9	57	N	78	c	99	x	120
DLE	16	%	37	:	58	O	79	d	100	y	121
DC1	17	&	38	;	59	P	80	e	101	z	122
DC2	18	'	39	<	60	Q	81	f	102	{	123
DC3	19	(	40	=	61	R	82	g	103	|	124
DC4	20	)	41	>	62	S	83	h	104	}	125

Πρόχειρο