

## ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΠΕΡΙΟΔΟΥ

Μάθημα: Προγραμματισμός,

AM:.....

Τμήμα: Ηλεκτρολόγων Μηχανικών, Μηχανικών Υπολογιστών

### ΘΕΜΑ 1

Αν η ακέραια μεταβλητή AM έχει δηλωθεί και η τιμή της είναι το αριθμητικό μέρος του αριθμού μητρώου σας, τι θα δείξουν οι συναρτήσεις printf(); (απαντήστε, χωρίς αιτιολόγηση, στο φύλλο των θεμάτων). [2μ]

(α) [0,5μ]

```
int i=10;
sum=AM-20000;
int *pi;
pi=&i;
do {
    sum +=*pi;
    i++;
} while (i<=15);
printf("sum=%d", sum);
```

(β) [0,5μ]

```
int a=0;
s=AM+100;
int P[ ]={9,1,3,3,4,0,1,2,3,4};
while (a<10) {
    if (a>5)
        s+=*(P+a);
    a++;
}
printf ("s=%d",s);
```

(γ) [0,5μ]

```
int i, j, M;
int R[2][3] ={{0,1,2}, {2,1,0}};
M = 0;
for (i=0; i < 2; i++) {
    for (j=0; j < 3; j++)
        if (R[i][j]< 2) M++;
}
AM - = M;
printf("AM= %d", AM);
```

(δ) [0,5μ]

```
int a, *pa;
a = AM+1;
for (pa=&a; a<AM+5;(*pa)++)
    printf("%d ", a%2);
AM=AM -a/20000;
printf ("\n AM=%d\n",AM);
```

### ΘΕΜΑ 2

Γράψτε τον ορισμό μιας συνάρτησης **read()**, η οποία θα δέχεται ως παραμέτρους (ορίσματα) 2 ακέραιους **i** και **j** και θα επιστρέφει:

- την τιμή  $2i+j$ , αν  $i>j$ ,
- την τιμή  $i2-2j$ , αν  $i<j$  ή
- την τιμή 0, αν  $i=j$ .

Στη συνέχεια, γράψτε τις εντολές για να δώσετε τιμές σ' ένα πίνακα **PP** πραγματικών αριθμών, 2 διαστάσεων  $M \times N$  καλώντας τη συνάρτηση **read()** και για να βρείτε τον μικρότερο αριθμό της 1<sup>ης</sup> στήλης και τον Μέσο Όρο της τελευταίας γραμμής του πίνακα. Θεωρείστε τα M και N σταθερές. [1,5μ]

### ΘΕΜΑ 3

Μια εγγραφή ενός μεταχειρισμένου αυτοκινήτου αποτελείται από τον αριθμό κυκλοφορίας (συμβολοσειρά), τα χιλιόμετρα που έχει κάνει (ακέραιος) και την τιμή πώλησης (πραγματικός). Δώστε τις εντολές που απαιτούνται για να διαβάσετε από το πληκτρολόγιο, **N** τέτοιες εγγραφές (αριθμός κυκλοφορίας κλπ.) και να τις αποθηκεύσετε σε ένα αρχείο κειμένου. Υποθέστε ότι το αρχείο κειμένου δεν υπάρχει και ότι θα πρέπει να δημιουργήσετε την κατάλληλη ροή, καθώς και να γίνουν οι απαραίτητοι έλεγχοι για τυχόν λάθη. Επίσης, υποθέστε ότι το **N** έχει δηλωθεί και ότι έχει κάποια τιμή (πχ = 5). [1μ]

#### ΘΕΜΑ 4

Γράψτε τις εντολές που απαιτούνται για να βρείτε τον μέσο όρο όλων των μονών αριθμών ενός πίνακα **N** ακεραίων **PA**, χρησιμοποιώντας τον όνομα του πίνακα σαν δείκτη (pointer) για να επεξεργαστείτε κάθε στοιχείο του – δηλαδή, να μην χρησιμοποιήσετε τον συμβολισμό **PA [i]** για την επεξεργασία των στοιχείων του πίνακα. Υποθέστε ότι ο πίνακας **PA[N]** έχει δηλωθεί και έχουν αποδοθεί τιμές στα στοιχεία του. [1μ]

#### ΘΕΜΑ 5

A. Γράψτε τις εντολές για να δηλώσετε μια δομή με όνομα **item**, η οποία θα περιλαμβάνει τον κωδικό ενός ανταλλακτικού, τη τιμή του, την ποσότητα που υπάρχει στη αποθήκη και μια άλλη δομή (υπο-δομή) η οποία θα περιέχει την ημερομηνία (έτος-μήνας- ημέρα) της παραγωγής του ανταλλακτικού. [0,5μ]

B. Στη συνέχεια γράψτε τον ορισμό μιας συνάρτησης **get\_item()** η οποία θα διαβάζει από το πληκτρολόγιο δεδομένα για να δίνει τιμές στα πεδία μιας δομής τύπου **item** και να επιστρέφει μια τέτοια δομή στο πρόγραμμα που την καλεί. [1μ]

Γ. Τέλος, γράψτε τις εντολές για να δηλώσετε ένα πίνακα **ITEMS**, **N** δομών τύπου **item** σε ένα κύριο πρόγραμμα, να δώστε τιμές στα στοιχεία όλων των δομών του πίνακα **ITEMS** με την βοήθεια της συνάρτησης **get\_item()** και να βρείτε ποιο ανταλλακτικό είναι το φτηνότερο. [1μ]

#### ΘΕΜΑ 6

Γράψτε τις εντολές με τις οποίες από ένα πίνακα 2 διαστάσεων (MxN) με όνομα **KAT** που περιέχει καταναλώσεις ηλεκτρικού ρεύματος, θα δημιουργούνται δύο άλλοι νέοι μονοδιάστατοι πίνακες, από τους οποίους, ο ένας **KMORE** θα περιέχει τις καταναλώσεις που είναι πάνω από το 400 kWh και ο άλλος **KLESS** τις καταναλώσεις που είναι κάτω από ή ίσες με 400 kWh. Βρείτε πόσες καταναλώσεις είναι κάτω από 100 kWh στον πίνακα **KLESS**. Επειδή δεν θα είναι γνωστό το μέγεθος τους, η δημιουργία των 2 νέων μονοδιάστατων πινάκων θα γίνεται δεσμεύοντας δυναμικά την απαιτούμενη μνήμη για τον καθένα. Υποθέστε ότι ο πίνακας **KAT** έχει δημιουργηθεί και έχει τιμές. [2μ] (αν δεν χρησιμοποιήσετε δυναμική δέσμευση μνήμης, η βαθμολογία της άσκησης θα είναι 1μ)

<code>FILE *fp;</code>
<code>FILE *fopen(char *fname, char *mode)</code> - Επιστρέφει είτε μια δομή τύπου <code>FILE</code> (επιτυχία) είτε <code>NULL</code> (αποτυχία)
<code>int fclose(FILE *fp)</code> - Επιστρέφει <code>0</code> (επιτυχία) είτε <code>EOF</code> (αποτυχία)
<code>int fputs(char *str, FILE *fp)</code> - Επιστρέφει μια μη αρνητική τιμή (επιτυχία) είτε <code>EOF</code> (αποτυχία)
<code>char *fgets(char *str, int num, FILE *fp)</code> - Επιστρέφει τον δείκτη <code>str</code> (επιτυχία) είτε <code>NULL</code> (αποτυχία)
<code>int fprintf(FILE *fp, char *format, ...)</code> - Επιστρέφει αριθμό χαρακτήρων που γράφτηκαν, είτε μια αρνητική τιμή (αποτυχία)
<code>int fscanf(FILE *fp, const char *format, ...)</code> - Επιστρέφει αριθμό των δεδομένων που διαβάστηκαν, είτε <code>NULL</code> (αποτυχία)
<code>int fread(void *buffer, size_t size, size_t num, FILE *fp)</code> - Επιστρέφει <code>num</code> (επιτυχία) είτε κάποιον άλλο αριθμό (αποτυχία)
<code>int fwrite(void *buffer, size_t size, size_t num, FILE *fp)</code> - Επιστρέφει <code>num</code> (επιτυχία) είτε κάποιον άλλο αριθμό (αποτυχία)
<code>int feof(FILE *fp)</code> - Επιστρέφει μια μη μηδενική τιμή (τέλος αρχείου) είτε <code>0</code> (δεν έχουμε φτάσει στο τέλος του αρχείου)
<code>int ferror(FILE *fp)</code> - Επιστρέφει <code>0</code> (τέλος αρχείου) είτε μη-μηδενική τιμή (σφάλμα)
<b>Πρότυπα συναρτήσεων της C σχετικά με τη δυναμική δέσμευση μνήμης</b>
<code>void *malloc(size_t size)</code> - επιστρέφει ένα δείκτη στη μνήμη που δεσμεύτηκε είτε <code>NULL</code> (αποτυχία)
<code>void *calloc(size_t num, size_t size)</code> - επιστρέφει ένα δείκτη στη μνήμη που δεσμεύτηκε, είτε <code>NULL</code> (αποτυχία)
<code>void *realloc(void ptr, size_t size)</code> - επιστρέφει ένα δείκτη στη μνήμη που δεσμεύτηκε, είτε <code>NULL</code> (αποτυχία)
<code>void free(void *ptr);</code>