

ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (C)

ΔΙΔΑΣΚΩΝ: Ν. ΒΑΣΙΛΑΣ

ΛΥΣΕΙΣ ΠΑΛΑΙΩΝ ΘΕΜΑΤΩΝ

Θέμα 1.

Αντικατάσταση MF1(a-b) με $((a-b > 0) ? a-b : -(a-b)) --> 5$

Αντικατάσταση MF2(a-b) με $(a-b > 0) ? a-b : -a-b --> -15$

MF1=5, MF2=-15

Θέμα 2.

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

```
int isvowel(char ch);
```

```
int main()
```

```
{
```

```
    int cnt=0;
```

```
    char ch;
```

```
    printf("Please enter a sentence\n");
```

```
    while ((ch = getchar()) != '\n')
```

```
        cnt += isvowel(ch);
```

```
    printf("There were %d vowels in the sentence.\n",cnt);
```

```
    return 0;
```

```
}
```

```
int isvowel(char ch) /* using a switch */
```

```
{
```

```
    switch (ch)
```

```
    {
```

```
        case 'a':
```

```
        case 'e':
```

```
        case 'i':
```

```
        case 'o':
```

```
        case 'u': return 1;
```

```
        default: return 0;
```

```
    }
```

```
}
```

Θέμα 3.

Αναδρομικός υπολογισμός παραγοντικού. **function(4) = 24, function(-2) = 0.**

Θέμα 4.

1. Ανοίγει το αρχείο κειμένου "pinakas.dat" για διάβασμα ελέγχοντας παράλληλα αν το άνοιξε χωρίς πρόβλημα (σε αντίθετη περίπτωση βγαίνει με την exit(-1)). 2. Διαβάζει ακέραιο στη μεταβλητή dim που αντιστοιχεί στο πλήθος των πραγματικών αριθμών που είναι αποθηκευμένοι στο αρχείο. 3. Δεσμεύει δυναμικά μνήμη και επιστρέφει τον δείκτη στον MAT. 4. Διαβάζει τους αριθμούς και τους βάζει στον πίνακα στον οποίο δείχνει ο MAT. 5. Κλείνει το αρχείο. 6. Εμφανίζει τα στοιχεία του πίνακα. 7. Αποδεσμεύει τη μνήμη.

Θέμα 5.

```
#include <stdio.h>

int removechar(char *str, char ch);

int main()
{
    int cnt=0;
    char ch, str[256];

    printf("Please enter a sentence:\n");
    gets(str);
    cnt = removechar(str, 'a');
    printf("There were %d a's.\n",cnt);
    printf("The new string is \"%s\".\n",str);
    return 0;
}

int removechar(char *str, char ch)
{
    int cnt=0, i=0;
    char *p;

    p = str;
    while (*p != '\0')
    {
        if (*p != ch)
            *(str+i++) = *p;    /* or str[i++] = *p; */
        else
            cnt++;
        p++;
    }
    *(str+i) = '\0';
    return cnt;
}
```

Θέμα 6.

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

```

int main()
{
    char name[256], am[10];
    FILE *fp;

    if ((fp = fopen("myname.txt","w")) == NULL)
    {
        printf("Can't open file \"myname.txt\" for writing.\n");
        exit(1);
    }
    printf("Please enter your full name: ");
    gets(name);
    printf("Now enter your A.M.: ");
    gets(am);
    fprintf(fp,"%s\n%s",name,am);
    fclose(fp);
    return 0;
}

```

Θέμα 7.

```

#include <stdio.h>

int main()
{
    int i, N;
    char **pin,line[256];
    FILE *fp;

    if ((fp = fopen("names.txt","r")) == NULL)
    {
        printf("Can't open file \"names.txt\" for reading.\n");
        exit(1);
    }
    fscanf(fp,"%d\n",&N);
    printf("N = %d\n");

    pin = (char **) malloc(N*sizeof(char *));

    for(i=0; i<N; i++)
    {
        fgets(line,256,fp); /* Η fgets κρατάει και τα newlines */
        pin[i] = (char *) malloc(strlen(line)+1);
        strcpy(pin[i],line);
    }

    fclose(fp);

    for(i=0; i<N; i++)
    {
        puts(pin[i]);
    }
}

```

```
    free(pin[i]);
}

free(pin);

return 0;
}
```

Θέμα 8.

Προσομοίωση:

```
i = 0, j = 10
i = 1, j = 9
i = 2, j = 8
i = 3, j = 7
i = 4, j = 6
i = 3, j = 5
i = 2, j = 4
i = 1, j = 3
} μέσα από την for

i = 0, j = 2
```

Θέμα 10 & 11.

```
#include <stdio.h>
#define ERROR -1

struct block{
    char shape[15];
    char color[15];
    float weight;
    float height;
} *blocks;

int N;

void sortblocks(char *s);

int main()
{
    int i;
    char junk[256];
    FILE *fp;

    if((fp=fopen("blocks.txt","r"))==NULL)
    {
        printf("Can't open file blocks.txt\n");
        exit(ERROR);
    }
}
```

```

fscanf(fp,"%s %s %d\n",junk,junk,&N);
    /* ή fscanf(fp,"%*s %*s %d\n",&N); */
fgets(junk,256,fp);
fgets(junk,256,fp);
blocks = (struct block *) malloc(N*sizeof(struct block));
for(i=0; i<N; i++)
fscanf(fp,"%s%s%f%f",blocks[i].shape,blocks[i].color,
        &blocks[i].weight,&blocks[i].height);
fclose(fp);
sortblocks("weight");
sortblocks("height");
free(blocks);
return 0;
}

void sortblocks(char *s)
{
    int i,j;
    struct block blk;

    if (!strcmp(s,"weight"))
    {
        for(i=0; i<N-1; i++)
            for(j=i; j<N; j++)
                if(blocks[i].weight < blocks[j].weight)
                {
                    blk=blocks[i];
                    blocks[i]=blocks[j];
                    blocks[j]=blk;
                }
    }
    else
        for(i=0; i<N-1; i++)
            for(j=i; j<N; j++)
                if(blocks[i].height < blocks[j].height)
                {
                    blk=blocks[i];
                    blocks[i]=blocks[j];
                    blocks[j]=blk;
                }
    for(i=0; i<N; i++)
        printf("%s \t%s\t%.1f\t%.1f\n",blocks[i].shape,
            blocks[i].color,blocks[i].weight,blocks[i].height);
}

```

Θέμα 12.

α) Τα μεγέθη σε bytes των παραπάνω τύπων δεδομένων είναι: 4 για (int), 2 για (short int), 1 για (char), 8 για (double) και 4 για οποιοδήποτε δείκτη (αφού αποθηκεύουμε μια διεύθυνση,

δηλαδή έναν μη προσημασμένο ακέραιο των 4 bytes για 32-μπιτα συστήματα). Ο πίνακας a χρειάζεται 10 θέσεις (short int) x 2bytes/(short int) = 20 bytes, ο δείκτης b 4 bytes και η μεταβλητή v όσο και το άθροισμα των bytes για τα πεδία της δομής, δηλαδή: 4 bytes για το πεδίο c, 20 θέσεις (char) x 1 byte/(char) = 20 bytes για το πεδίο του πίνακα χαρακτήρων s1, 4 bytes για το πεδίο του δείκτη s2 και 8 bytes για το πεδίο του x, σύνολο 36 bytes. Συνεπώς, η printf θα εμφανίσει:

```
20
4
36
```

β) Στην πρώτη επανάληψη, $i=-2$, $a+i=5-2=3$, $\text{func}(3)=y=1+3=4$ (static), $a=4\%5=4$. Στη δεύτερη επανάληψη, $i=-2+3=1$, $a+i=4+1=5$, $\text{func}(5)=y=4+5=9$ (static), $a=9\%5=4$. Τέλος, στην τρίτη επανάληψη, $i=1+3=4$, $a+i=4+4=8$, $\text{func}(8)=y=9+8=17$ (static), $a=17\%5=2$. Μετά το i γίνεται 7 και εξέρχεται από τον βρόχο for. Συνεπώς, η printf θα εμφανίσει:

```
a = 4
a = 4
a = 2
```

γ) Η διακήρυξη [int x=1, y=0;] της main() δημιουργεί δύο τοπικές μεταβλητές με ίδια ονόματα με αυτά των καθολικών μεταβλητών x και y. Η μόνη καθολική μεταβλητή που «βλέπει» η main είναι η a. Αντιθέτως, οι τρεις συναρτήσεις func1, func2 και func3 «βλέπουν» και τις τρεις καθολικές μεταβλητές. Με την κλήση της συνάρτησης func1 εκχωρούνται οι τιμές 10 και 5 στις καθολικές μεταβλητές x και y αντίστοιχα. Με την κλήση της func3 με ορίσματα τις τοπικές μεταβλητές της main (x=1 και y=0) θα έχουμε ότι $a = \text{func}(1,0) = 1-0 = 1$. Στη συνέχεια, η κλήση της func3 μεταβάλλει την τιμή της καθολικής μεταβλητής a στο 0 και εμφανίζει στην οθόνη τις καθολικές μεταβλητές x και y. Κατά συνέπεια, η έξοδος του προγράμματος θα είναι η εξής:

```
x = 10 y = 5
a = 0
```

Θέμα 13.

```
char *cleanstr(char *instr)
{
    int i=0;
    char *outstr;

    ostr = (char *) malloc(strlen(instr)+1);

    while(*instr != '\0')
    {
        if(*instr < '0' || *instr > '9')
            ostr[i++] = *instr;

        instr++;
    }
    ostr[i] = '\0';
    return ostr;
}
```

Θέμα 14.

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

char *cleanstr(char *);

int main()
{
    char line[256];
    FILE *fp;

    if((fp = fopen("text1.txt", "r")) == NULL)
    {
        printf("Can't open file text1.txt. Exiting ...\n");
        exit(-1);
    }
    while(!feof(fp))
        if (fgets(line,256,fp) != NULL)
            /* ή απλά: if(fgets(line,256,fp) */
            printf("%s",cleanstr(line));

    fclose(fp);
    return 0;
}
```

Θέμα 15.

```
#include <stdio.h>
int main()
{
    int i,N;
    float *vathmoi,mo,min,max;
    FILE *fp;

    if((fp = fopen("C_grades.txt","r")) == NULL)
    {
        printf("Can't open file \"C_grades.txt\" for reading.\n");
        exit(-1);
    }
    fscanf(fp,"%d\n",&N);

    vathmoi = (float *)malloc(N*sizeof(float));

    for(i=0; i<N; i++)
        fscanf(fp,"%f",&vathmoi[i]);
    fclose(fp);

    mo = min = max = vathmoi[0];
    for(i=1; i<N; i++)
```

```

{
    if(vathmoi[i] < min)
        min = vathmoi[i];
    if(vathmoi[i] > max)
        max = vathmoi[i];
    mo += vathmoi[i];
}
mo = mo/N;

printf("Megistos: %.1f\n",max);
printf("Elaxistos: %.1f\n",min);
printf("Mesos oros: %.1f\n",mo);

free(vathmoi);

return 0;
}

```

Θέμα 16.

```

char  gamma(int pos, int cap)
{
    if((pos < 0) || (pos > 26) || ((cap != 0) && (cap != 1)))
        return '\0';

    return ((cap == 1) ? pos+64 : pos+96);
}

```