

Εξεταζόμενο μάθημα : Προγραμματισμός Ι

Όνοματεπώνυμο σπουδαστή/άστριας : .....

Εξάμηνο σπουδών : ..... Α.Ε.Μ. : .....

Ιούνιος 2007

### ΔΙΑΡΚΕΙΑ ΕΞΕΤΑΣΗΣ : 2 ΩΡΕΣ

Οι εξεταζόμενοι επιτρέπεται να χρησιμοποιήσουν παντός είδους έντυπα βοηθήματα.

Τα θέματα επισυνάπτονται στο γραπτό με συμπληρωμένο το πλαίσιο κορυφής.

#### Θέμα 1<sup>ο</sup> (35%)

Να σχεδιαστεί και να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού C, το οποίο θα επιτελεί τα ακόλουθα :

- Θα δημιουργεί ένα πίνακα αλφαριθμητικών 2 X 50 χαρακτήρων.
- Θα διαβάζει δύο αλφαριθμητικά τα οποία και θα αποθηκεύει στον πίνακα.
- Θα υπολογίζει το μήκος των δύο αλφαριθμητικών χωρίς την χρήση της συνάρτησης `strlen()`. Τα αρχικά αλφαριθμητικά και τα μήκη αυτών θα εμφανίζονται στην οθόνη.
- Θα αντιγράφει τους χαρακτήρες από θέση  $\alpha$  έως θέση  $\beta$  του πρώτου αλφαριθμητικού στο τέλος του δευτέρου και θα το αποθηκεύει στον νέο πίνακα αλφαριθμητικών `final[50]`.
- Οι θέσεις  $\alpha$ ,  $\beta$  θα εισάγονται από τον χρήστη και θα γίνεται έλεγχος ορθότητας.
- Να εμφανίζει το νέο αλφαριθμητικό καθώς και τα μήκος αυτού.

Δίνεται ότι :

- Τα αλφαριθμητικά δόθηκαν σωστά, με κεφαλαία λατινικά γράμματα, και δεν απαιτείται έλεγχος γι' αυτό

```

#include <stdio.h>
#include <conio.h>

main()
{
    char names[2][50];
    char final_name[50];
    int size0,size1,final_size, start, end,i,j;

    // Eisagwgh arxikwn string
    printf("Give the first name :");
    gets(names[0]);
    printf("Give the second name :");
    gets(names[1]);
    // Emfanish & Ypologismos megethous 1ou string
    printf("\nfirst string : %s",names[0]);
    size0=0;
    while(names[0][size0] != '\0')
    {
        size0++;
    }
    printf("\t megethos = %d",size0);
    // Emfanish & ypologismos megethous 2ou string
    printf("\nsecond string : %s",names[1]);
    size1=0;
    while(names[1][size1] != '\0')
    {
        size1++;
    }
    printf("\t megethos = %d",size1);
    // Eisagwgh arxikhs theshs antigrafhs & elegxos egkyrothtas
    printf("\nGive start copy position of the first string:");
    scanf("%d",&start);
    while(start >size0)
    { printf("error.....");
      printf("\nthe start copy position must be smaller than the size of the string (%d)",size0);
      printf("\nGive start copy position of the first string again:");
      scanf("%d",&start);
    }
    // Eisagwgh telikhs theshs antigrafhs & elegxos egkyrothtas
    printf("\nGive end copy position of the first string:");
    scanf("%d",&end);
    while((end > size0) || (end <start))
    { printf("error.....");
      printf("\nthe end copy position must be smaller than the size of the string (%d) and bigger than
the start position (%d)",size0, start);
      printf("\nGive end copy position of the first string again:");
      scanf("%d",&end);
    }
    //Dhmiourgia neou alfarithmhtikou
    for(j=0;j<=size1;j++)
    {
        final_name[j] = names[1][j];
    }
    for(i=start;i<=end;i++)
    {
        final_name[j-1] = names[0][i];
        j++;
    }
    final_name[j-1] = '\0';
    printf("Final string %s",final_name);
    /* final_size=0;

```

```

while(final_name[final_size] != '\0')
{
    final_size++;
}*/
printf("\t Size of final string = %d",j-1);
getch();
}

```

### Θέμα 2<sup>ο</sup> (45%)

Στα πλαίσια ανταγωνισμού εταιριών κινητής τηλεφωνίας, η εταιρεία “X” θέλει να υλοποιήσει καλοκαιρινές προσφορές στα τιμολόγια πελατών. Πριν όμως την δημοσίευση των προσφορών καλείται ένας μηχανικός λογισμικού να υλοποιήσει μια μελέτη συμφερόντων. Η μελέτη υλοποιείται στους χρόνους ομιλίας 12 μηνών σε λεπτά για ένα δείγμα πελατών (πλήθος δέκα). Έχοντας υπόψη και έναν πίνακα ο οποίος περιέχει τις κατηγορίες προγραμμάτων των πελατών (τιμές 1-5). Να σχεδιαστεί και να υλοποιηθεί πρόγραμμα H/Y σε γλώσσα προγραμματισμού C το οποίο :

1. Να δηλώνει τους πίνακες και να τοποθετεί αρχικές τιμές δια μέσω του προγράμματος
2. Να εμφανίζει την κατηγορία προγράμματος του πελάτη, τους χρόνους ομιλίας αυτού για κάθε μήνα και τον συνολικό ετήσιο χρόνο ομιλίας.
3. Να υπολογίζει το κόστος κλήσεων για κάθε μήνα (0,234 Cents/λεπτό). Προσοχή!. Να αφαιρείται ο δωρεάν χρόνος ομιλίας.
4. Να εμφανίζει για κάθε πελάτη το εάν συμφέρει να ανέβει κατηγορία προγράμματος. Σημείωση. Αλλαγή κατηγορίας γίνεται όταν ο μέσος μηνιαίος χρόνος ομιλίας είναι 1 ½ φορά παραπάνω από τον δωρεάν χρόνο ομιλίας της κατηγορίας του. Παράδειγμα - Πελάτης στην κατηγορία 2 με μέσο ετήσιο χρόνο ομιλίας 185 συμφέρει να ανέβει κατηγορία.
5. Να εμφανίζει το συνολικό ποσοστό πελατών που τους συμφέρει να ανέβουν κατηγορία προγράμματος

1	2	3	4	5
Free Time 60	Free Time 120	Free Time 240	Free Time 360	Free Time 600

```

#include <stdio.h>
#include <conio.h>
#define n 3 // Example for 3 customers, must change to 10
#define m 12

main()
{
    int times[n][m]={ {400, 300, 600, 200, 150, 240, 70, 300, 900, 600, 500, 440},
                    {120, 150, 100, 80, 100, 40, 60, 35, 95, 200,120, 48},
                    {35, 45, 66, 90, 80, 240, 20, 20, 20, 40, 50, 44}};

    int categories[n]= {2,3,1}, total_talk_time[n];
    int i,j;
    float count=0;

// Read Data From keyboard
/* for(i=0;i<n;i++)
{ printf("\nCustomer No %d",i+1);
  printf("\n\t Categorie program :");
  scanf("%d",&categories[i]);
  for(j=0; j<m; j++)
  { printf("\n\tTime talk for month no %d (seconds) :",j+1);
    scanf("%d",&times[i][j]);
  }
}
*/

// Print Customer talk time and calculate total talk time
for(i=0;i<n;i++)
{ printf("\nCustomer No %d",i+1);
  printf("\n\t Categorie program : %d \n",categories[i]);
  printf(" 1 2 3 4 5 6 7 8 9 10 11 12\n");
  for(j=0; j<m; j++)
  { printf("%3d ",times[i][j]);
    total_talk_time[i] += times[i][j];
  }
}

// Calculate invoice charge

printf("\n K O S T O L O G I S H");
for(i=0;i<n;i++)
{ printf("\nCustomer No %d",i+1);
  printf("\n\t Categorie program : %d \n",categories[i]);
  printf("\n\t Monthly Charge\t\t");
  printf(" 1 2 3 4 5 6 7 8 9 10 11 12\n");
  for(j=0; j<m; j++)
  {
    switch(categories[i])
    { case 1 : if( (times[i][j]-60) > 0)
              printf("%.2f ",(times[i][j]-60) * 0.234);
            else
              printf("0 ");
            break;
      case 2 : if( (times[i][j]-120) > 0)
              printf("%.2f ",(times[i][j]-120) * 0.234);
            else
              printf("0 ");
            break;
      case 3 : if( (times[i][j]-240) > 0)
              printf("%.2f ",(times[i][j]-240) * 0.234);
            else

```

```

        printf("0 ");
        break;
    case 4 : if( (times[i][j]-360) > 0)
        printf("%.2f ",(times[i][j]-360) * 0.234);
        else
            printf("0 ");
            break;
    case 5 : if( (times[i][j]-600) > 0)
        printf("%.2f ",(times[i][j]-600) * 0.234);
        else
            printf("0 ");
    }
}
}

```

// Found change talk program

```

count = 0;
for(i=0;i<n;i++)
{
    switch(categories[i])
    { case 1 : if( (total_talk_time[i]/12 >= 60*1.5))
        {
            printf("\nCustomer %d has a benefit to go to 2nd program ",i+1);
            count++;
        }
        else
            printf("\nCustomer %d should stay to the same talk program ", i+1);
            break;
    case 2 : if( (total_talk_time[i]/12 >= 120*1.5))
        {
            printf("\nCustomer %d has a benefit to go to 3rd program ",i+1);
            count++;
        }
        else
            printf("\nCustomer %d should stay to the same talk program ",i+1);
            break;
    case 3 : if( (total_talk_time[i]/12 >= 240*1.5))
        {
            printf("\nCustomer %d has a benefit to go to 4th program ",i+1);
            count++;
        }
        else
            printf("\nCustomer %d should stay to the same talk program ",i+1);
            break;
    case 4 : if( (total_talk_time[i]/12 >= 360*1.5))
        {
            printf("\nCustomer %d has a benefit to go to 5th program ",i+1);
            count++;
        }
        else
            printf("\nCustomer %d should stay to the same talk program ",i+1);
            break;
    case 5 : if( (total_talk_time[i]/12 >= 600*1.5))
        {
            printf("\nCustomer %d has a benefit to go to 6th program ",i+1);
            count++;
        }
        else
            printf("\nCustomer %d should stay to the same talk program ",i+1);
            break;
    }
}
}

```

```
printf("\n Pososto Epitychias programmatos = %f", (count/n)*100);  
getch();  
}
```

### Θέμα 3<sup>ο</sup> (20%)

Για το πρόγραμμα που ακολουθεί εντοπίστε τα δέκα συντακτικά λάθη αυτού και συμπληρώστε τις επιλογές του μενού. Δύο εντολές του προγράμματος είναι λογικά εσφαλμένες. Εντοπίστε αυτές και προτείνετε την αντικατάστασή τους . Τι υλοποιεί το πρόγραμμα; (3 γραμμές).

Πρόγραμμα Θέματος Νο3.

```
#include <stdio.h>
#include <conio.h>
define n 3

main()
{
    int arr1[n][n] = { {1,2,3},{4,5,6},{7,8,9}}
    arr2[n][n] = { {9,8,7},{6,5,4},{3,2,1}};
    arr3[n][n], arr4[n][n];
    int i,j,k, epilogi;
    int flag1=0,flag2=0;

    epilogi = 5;
    while((epilogi<=5) OR (epilogi>=1))
    {
        clrscr();
        printf("\n +----- MENU -----+\n");
        printf("\n | 1.                               |\n");
        printf("\n | 2.                               |\n");
        printf("\n | 3.                               |\n");
        printf("\n | 4.                               |\n");
        printf("\n | 5.                               |\n");
        printf("\n | 6. Exit                          |\n");
        printf("\n +-----+\n");
        printf("\n   Epilogh :");
        scanf("%d",&epilogi);
        switch(epilogi)
        {
            case 1 : for(i=0;i<n;i++)
                    for(j=0;j<n;j++)
                    {
                        printf("\n arr1[%d][%d]: ",i+1,j+1);
                        scanf("%d",arr1[i][j]);
                    }
                    for(i=0;i<n;i++)
                    for(j=0;j<n;j++)
                    {
                        printf("\n arr2[%d][%d]: ",i+1,j+1);
                        scanf("%d",arr2[i][j]);
                    }

            case 2 : for(i=0;i<n;i++)
                    for(j=0;j<n;j++)
                    {
                        arr3[j][i] = arr1[j][i] + arr2[j][i];
                    }
                    printf("\n The calculation has been done...
Press any key");
                    getch();
                    flag1=1;
                    break;

            case 3 : for(i=0;i<n;i++)
                    for(j=0;j<n;j++)
                    {
                        arr4[j][i] = arr1[j][i] - arr2[j][i];
                    }

```

```

                    printf("\n The calculation has been done...
Press any key");
                    getch();
                    flag2=1;
                    break;

            case 4 : if(flag1 AND flag2)
                    { printf("ARRAY 1\t\t ARRAY2\n");
                    for(i=0;i<n;i++)
                    { printf("\n");
                    for(j=0;j<n;j++)
                    {
                        printf("%d ",arr1[i][j]);
                    }
                    printf("\t\t");
                    for(j=0;j<n;j++)
                    {
                        printf("%d ",arr2[i][j]);
                    }
                    printf("\t\t");
                    for(j=0;j<n;j++)
                    {
                        printf("%d ",arr3[i][j]);
                    }
                    printf("\t\t");
                    for(j=0;j<n;j++)
                    {
                        printf("%d ",arr4[i][j]);
                    }
                    }
                    }
                    else
                    printf("error... Please perform both
calculations first");
                    getch();
                    break;

            case 5 : printf("ARRAY 1\t\t ARRAY2\n");
                    for(i=0;i<n;i++)
                    { printf("\n");
                    for(j=0;j<n;j++)
                    {
                        printf("%d ",arr1[i][j]);
                    }
                    printf("\t\t");
                    for(j=0;j<n;j++)
                    {
                        printf("%d ",arr2[i][j]);
                    }
                    }
                    }
                    getch();
                    break;

            case 6 : exit();
                    break;

            default : printf("\n Error... Please try again);
                    getch();
                    }
                    }
                    }

```

```

#include <stdio.h>
#include <conio.h>
#define n 3

main()
{
    int arr1[n][n] = { {1,2,3},{4,5,6},{7,8,9}},
        arr2[n][n] = { {9,8,7},{6,5,4},{3,2,1}}, arr3[n][n], arr4[n][n];
    int i,j,k, epilogi;
    int flag1=0,flag2=0;

    epilogi = 5;
    while((epilogi<=5) || (epilogi>=1))
    {
        clrscr();
        printf("\n +----- M E N U -----+\n");
        printf("\n | 1. Read Array Values      |\n");
        printf("\n | 2. Add Arrays              |\n");
        printf("\n | 3. Subtract Arrays         |\n");
        printf("\n | 4. Display Arrays and calculations |\n");
        printf("\n | 5. Display Initial arrays  |\n");
        printf("\n | 6. Exit                    |\n");
        printf("\n +-----+\n");
        printf("\n   Epilogh :");
        scanf("%d",&epilogi);
        switch(epilogi)
        {
            case 1 : for(i=0;i<n;i++)
                    for(j=0;j<n;j++)
                    {
                        printf("\n arr1[%d][%d]: ",i+1,j+1);
                        scanf("%d",&arr1[i][j]);
                    }
                    for(i=0;i<n;i++)
                    for(j=0;j<n;j++)
                    {
                        printf("\n arr2[%d][%d]: ",i+1,j+1);
                        scanf("%d",&arr2[i][j]);
                    }
                    break;
            case 2 : for(i=0;i<n;i++)
                    for(j=0;j<n;j++)
                    {
                        arr3[j][i] = arr1[j][i] + arr2[j][i];
                    }
                    printf("\n The calculation has been done... Press any key");
                    getch();
                    flag1=1;
                    break;
            case 3 : for(i=0;i<n;i++)
                    for(j=0;j<n;j++)
                    {
                        arr4[j][i] = arr1[j][i] - arr2[j][i];
                    }
                    printf("\n The calculation has been done... Press any key");
                    getch();
                    flag2=1;
                    break;
            case 4 : if(flag1 && flag2)

```



```

{ printf("ARRAY 1\t\t ARRAY2\n");
  for(i=0;i<n;i++)
  { printf("\n");
    for(j=0;j<n;j++)
    {
      printf("%d ",arr1[i][j]);
    }
    printf("\t\t");
    for(j=0;j<n;j++)
    {
      printf("%d ",arr2[i][j]);
    }
    printf("\t\t");
    for(j=0;j<n;j++)
    {
      printf("%d ",arr3[i][j]);
    }
    printf("\t\t");
    for(j=0;j<n;j++)
    {
      printf("%d ",arr4[i][j]);
    }
  }
}
else
  printf("error... Please perform both calculations first");
getch();
break;
case 5 : printf("ARRAY 1\t\t ARRAY2\n");
for(i=0;i<n;i++)
{ printf("\n");
  for(j=0;j<n;j++)
  {
    printf("%d ",arr1[i][j]);
  }
  printf("\t\t");
  for(j=0;j<n;j++)
  {
    printf("%d ",arr2[i][j]);
  }
}
getch();
break;
case 6 : exit();
break;
default : printf("\n Error... Please try again");
getch();
}
}
}
}
}

```