Πανεπιστήμιο Κρήτης, Τμήμα Εφαρμοσμένων Μαθηματικών,

Χ.Ε. 2009, ΕΜ 101 Εισαγωγή στους Η/Υ (Π. Λεκέας)

Τελική Εξέταση, Ιανουάριος 2010

Το συγκεκριμένο διαγώνισμα έχει άριστα το 100. Θυμηθείτε επίσης ότι το διαγώνισμα αυτό μετράει σε ποσοστό 70% του τελικού βαθμού του μαθήματος ΤΕΜ 101 "Εισαγωγή στους Η/Υ" (το άλλο 30% προέρχεται από τα εργαστήρια). Για να θεωρηθεί ότι περάσατε το μάθημα πρέπει από το σημερινό διαγώνισμα να μαζέψετε τουλάχιστον 50 μονάδες. Καλή Επιτυχία !

Άσκηση 1, (μον. 24): Απαντήστε στις ακόλουθες ερωτήσεις:

(Α1Ε1, μον. 8) Τί είναι η διερμηνεία; Ποιά η διαφορά της από τη μεταγλώττιση;

(Α1Ε2, μον. 8) Τί είναι ένα σημασιολογικό λάθος; Σε τί διαφέρει από ένα συντακτικό λάθος; Με ποιόν τρόπο ανιχνεύονται τα πιο πάνω λάθη;

(Α1Ε3, μον. 8) Δώστε τον ορισμό της δομής δεδομένων δένδρου. Είναι ένα δένδρο μία αναδρομική δομή δεδομένων; Γιατί;

Απάντηση: Α1Ε1: 225, Α1Ε2: 40, Α1Ε3: 83 (οι σελίδες αναφέρονται στο βιβλίο "Εισαγ. στη Συγχρ. Επιστ. των Η/Υ")

Άσκηση 2, (μον. 12): Απαντήσετε στις εξής ερωτήσεις σωστού ή λάθους (η κάθε ερώτηση μετρά 3 μονάδες. **Προσοχή !! υπάρχει αρνητική βαθμολόγηση σε αυτή την άσκηση. Κάθε λάθος απάντηση αφαιρεί 3 βαθμούς από το σύνολο !!**).

1. Στο UNIX η διαταγή **mdir askiseis** δημιουργεί στον τρέχον κατάλογο ένα καινούργιο κατάλογο με όνομα askiseis.

2. Στο UNIX η διαταγή **rm askisei*** διαγράφει από τον τρέχον κατάλογο όλους τους καταλόγους που υπάρχουν μέσα, που η ονομασία τους ξεκινά από "askisei", εάν αυτοί είναι κενοί.

3. Στο UNIX η διαταγή **cp askisi.c askisi2.c** αντιγράφει το αρχείο askisi.c σε ένα άλλο αρχείο με όνομα askisi2.c στον ίδιο κατάλογο,

4. Στο UNIX η εντολή **cat askisi1.c** εμφανίζει το περιεχόμενο του αρχείου askisi1.c σε μία συνεχή ροή στη οθόνη του Η-Υ

Απάντηση: 1. Λ, 2. Λ, 3. Σ, 4. Σ

Άσκηση 3, (μον. 64): Απαντήστε στις ακόλουθες ερωτήσεις σχετικά με τα προγράμματα που σας δίνονται παρακάτω στην εκφώνηση.

(Α3Ε1, μον. 12) Τί θα τυπωθεί στην οθόνη, όταν εκτελεστεί το Πρόγραμμα 1;

(Α3Ε2, μον. 12) Τί θα τυπώνει το Πρόγραμμα 1, εάν αλλάξω στο define το n από 9 σε 10;

(Α3Ε3, μον. 12) Τί θα τυπωθεί από το Πρόγραμμα 1, εάν αλλάξω στο define το n από 9 σε 13 και το βήμα στην for το κάνω i = - 3 ;

(Α3Ε4, μον. 10) Το Πρόγραμμα 2 προήλθε από το Πρόγραμμα 1, προσθέτοντας μία συνάρτηση frac και προσθέτοντας άλλη μία εντολή printf μετά την τελευταία εντολή printf του Προγράμματος 1. Τί θα τυπώσει το Πρόγραμμα 2 εάν εκτελεστεί;

(Α3Ε5, μον. 10) Πόσες κλήσεις θα γίνουν από το Πρόγραμμα 2 στην frac εάν στο define αλλάξω το n από 9 σε 15;

(Α3Ε6, μον. 8) Τί θα γίνει εάν στην define του Προγράμματος 2 αλλάξω το n από 9 σε 16;

Πρόγραμμα 1, Άσκησης 3

```
#include <stdio.h>
#define n 9
```

```
int i,parity;
main(){
        for (i=n ; i>1 ; i--){
                parity=i%2;
                switch (parity) {
                case 0:
                        i++;
                        break;
                default:
                        i--;
                        break;
                }
                printf("%d\n",i);
}}
```

Πρόγραμμα 1, Άσκησης 3

Πρόγραμμα 2, Άσκησης 3

```
#include <stdio.h>
#define n 9
int i,parity;
int frac(int x,int y);
main(){
        for (i=n ; i>1 ; i--){
                parity=i%2;
                switch (parity) {
                case 0:
                        i++;
                        break;
                default:
                        i--;
                        break;
                }
                printf("%d\n",i);
                printf("klasma=%d\n",frac(parity,i));
}}
int frac(x,y){
x=y/x+y;
return(x); }
```

Πρόγραμμα 2, Άσκησης 3

Απάντηση:

A3E1:

```
8
6
4
2
```

A3E2: Θα τυπώνει συνέχεια 11

```
11
11
11
11
.
.
.
```

A3E3:

```
12
```

A3E4:

```
8
klasma=16
6
klasma=12
4
klasma=8
2
klasma=4
```

A3E5:

```
7
```

A3E6: Θα τυπωθεί πρώτα ο αριθμός 17 και μετά (στην πρώτη κλήση της frac) θα έχω διαίρεση με το μηδέν, οπότε ο H-Y θα σταματήσει τυπώνοντας το μύνημα "Floating exception". Δηλαδή αυτό που θα τυπωθεί θα είναι:

```
17
Floating exception
```