ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΠΛ131 Αρχές Προγραμματισμού

Ακαδημαϊκό Έτος 2016/17 – Χειμερινό Εξάμηνο

ΕΝΔΙΑΜΕΣΗ ΕΞΕΤΑΣΗ

ΗΜΕΡΟΜΗΝΙΑ: 29 Οκτωβρίου 2016
ΔΙΑΡΚΕΙΑ: 10:00πμ – 12:30μμ
ΑΙΘΟΥΣΕΣ: Κτήριο ΧΩΔ01, Αίθουσες 108 και 109
ΔΙΔΑΣΚΟΥΣΑ: Ελπίδα Κεραυνού-Παπαηλιού

Απαντήστε όλες τις ερωτήσεις
Κάθε ερώτηση βαθμολογείται με 25 μονάδες.

Ερώτηση 1

Για τις ακόλουθες περιπτώσεις κωδίκων σε Java, εξηγήστε κατά πόσον ο κώδικας (i) είναι ή δεν είναι λογικά ισοδύναμος με τον κώδικα (ii), και γιατί, παρουσιάζοντας τα αποτελέσματα της εκτέλεσης του κάθε κώδικα. Θεωρείστε ότι η συνάρτηση putchar που χρησιμοποιείται σε κάποιες από τις περιπτώσεις ορίζεται ως εξής:

```
public static void putchar (char c){
  System.out.print(c);}
```

Αρ.	Κώδικας (i)	Κώδικας (ii)

1.	`int n = -1;`
	`while (n > 0){`
	`  putchar('*');`
	`  n = n - 1;`
	`};`
		`int n = -1;`
		`do {`
		`  putchar('*');`
		`  n = n - 1;`
		`} while (n > 0);`

2.	`int x;`
	`do{`
	`  System.out.print(`
	`  "\nEnter num: ");`
	`  x=StdIn.readInt();`
		`int x;`
		`System.out.print("\nEnter num: ");`
		`x=StdIn.readInt();`
		`while(x <= 0){`
		`  System.out.print(`

	`} while (x <= 0);`	`"\nEnter num: ");`
		`x=StdIn.readInt();`
		`}`

3.	`int x = 5;`
	`float y = x / x * x;`
	`float z = (float) x /(x * x);`
		`int x = 5;`
		`float y = x / (x * x);`
		`float z = (float) (x /(x * x));`

4.	`int x = 10, res = 1;`
	`while (x > 1)`
	`  res = res * x;`
	`  x = x - 1;`
		`int x = 10; res = 1;`
		`do {`
		`  res = res * x;`
		`  x = x - 1;`
		`} while (x > 1);`

5.	`int y = 2016;`
	`boolean leap_y;`
	
	`if ((y % 4 == 0) &&`
	`  (y % 100 != 0 ||`
	`  y % 400 == 0))`
	`  leap_y = true;`
	`else leap_y = false;`
		`int y = 2016;`
		`boolean leap_y;`
		
		`if (y % 4 != 0) leap_y = false;`
		`else if (y % 100 != 0)`
		`  leap_y = true;`
		`else if (y % 400 == 0)`
		`  leap_y = true;`
		`else leap_y = false;`

6.	`int s = 5, r = 1, c;`
	
	`while (r <= s){`
	`  putchar('\n');`
	`  c = 1;`
	`  while (c <= s){`
	`  putchar('*');`
	`  c = c + 1;}`
	`  r = r + 1; }`
		`int s = 5, r = s, c;`
		
		`while (r >= 1){`
		`  putchar('\n');`
		`  c = s;`
		`  while (c >= 1){`
		`  putchar('*');`
		`  c = c - 1;}`
		`  r = r - 1; }`

7.	`int s = 5, r = 1, c;`
	
	`while (r <= s){`
	`  putchar('\n');`
	`  c = 1;`
	`  while (c <= r){`
	`  putchar('*');`
	`  c = c + 1;}`
	`  r = r + 1; }`
		`int s = 5, r = 1, c = 1;`
		
		`while (r <= s){`
		`  putchar('\n');`
		`  while (c <= r){`
		`  putchar('*');`
		`  c = c + 1;}`
		`  r = r + 1; }`

8.	`int[] x = {5,10};`
	
	`public static void foo (int n,`
	`  int m){`
	`  int temp = n;`
	`  n = m;`
	`  m = temp;}`
	
	`foo(x[0],x[1]);`
		`int[] x = {5,10};`
		
		`public static void foo (int[] t){`
		`  if(t.length >= 2){`
		`  int temp = t[0];`
		`  t[0] = t[1];`
		`  t[1] = temp;`
		`  }`
		`}`
		
		`foo(x);`

Ερώτηση 2

(α) Κατασκευάστε πρόγραμμα σε Java για τον υπολογισμό της σειράς

$$1 - (x^2 / 2!) + (x^4 / 4!) - (x^6 / 6!) \ldots\ldots$$

για 15 όρους και δεδομένο φυσικό αριθμό x, ο οποίος εισάγεται από τη γραμμή εντολής. Οι συναρτήσεις για το παραγοντικό (!) και τη δύναμη ορίζονται ως ακολούθως:

```
\* Συνάρτηση για παραγοντικό *\
public static int factorial (int x) { int res = 1;
   while( x > 1){res = res * x; x = x-1;}
   return res;
 }

\* Συνάρτηση για δύναμη *\
public static int power (int x, int p) { int res = 1;
   while (p > 0) {res = res * x; p = p - 1;}
   return res;
  }
```

Πιο κάτω δίνονται κάποια παραδείγματα χρήσης του προγράμματος Series.java:

```
$ java Series 4
The result is: -0.86064

$ java Series 23
The result is: -169791.204876

$ java Series 9
The result is: 780.550632
```

(β) Κατασκευάστε πρόγραμμα σε Java το οποίο απλά παρουσιάζει σε ανάστροφη σειρά τα ορίσματα που λαμβάνει στη γραμμή εντολής. Η λειτουργία του εν λόγω προγράμματος (ReverseArgs.java) διαφαίνεται πιο κάτω:

```
$ java ReverseArgs Mon Tue Wed Thu Fri Sat Sun
Sun  Sat  Fri  Thu  Wed  Tue  Mon

$ java ReverseArgs
Nothing to reverse ...

$ java ReverseArgs 45 123 78 54
54  78  123  45

$ java ReverseArgs Today is Saturday October 29th 2016
2016  29th  October  Saturday  is  Today
```

Ερώτηση 3

Κατασκευάστε πρόγραμμα σε Java το οποίο μετατρέπει ένα θετικό ακέραιο αριθμό εκφρασμένο στη βάση n, όπου $2 \le n \le 9$ στη δεκαδική βάση. Για παράδειγμα, ο αριθμός 1101101 στη βάση 2 (δυαδικός αριθμός) ισοδυναμεί με το δεκαδικό αριθμό 109 ($= 2^0 + 2^2 + 2^3 + 2^5 + 2^6$). Όπως γνωρίζουμε το πεδίο των ψηφίων ενός δεκαδικού αριθμού είναι τα ψηφία 0, 1, .., 9, και ενός δυαδικού αριθμού είναι τα ψηφία 0 και 1. Παρομοίως, το πεδίο των ψηφίων ενός αριθμού στη βάση 5 είναι τα ψηφία 0, 1, .., 4. Ο αριθμός 12334 στη βάση 5 ισοδυναμεί με το δεκαδικό αριθμό 969 ($= 5^0 \times 4 + 5^1 \times 3 + 5^2 \times 3 + 5^3 \times 2 + 5^4 \times 1$), δηλαδή $12334_5 \equiv 969_{10}$. Πιο γενικά ένας αριθμός στη βάση n, όπου $2 \le n \le 9$, μπορεί να περιλαμβάνει μόνο ψηφία από το 0 μέχρι το $n - 1$. Έστω ο αριθμός A στη βάση n (A_n) απαρτίζεται από την ακολουθία ψηφίων

$$\Psi_m \Psi_{m-1} \ldots\ldots \Psi_2 \Psi_1 \Psi_0$$

όπου $0 \le \Psi_i < n$. Ο ισοδύναμος δεκαδικός αριθμός δίνεται από τον τύπο

$$n^m \times \Psi_m + n^{m-1} \times \Psi_{m-1} + \ldots\ldots + n^2 \times \Psi_2 + n^1 \times \Psi_1 + n^0 \times \Psi_0$$

Το πρόγραμμα (BaseConvert.java) λαμβάνει μια έγκυρη βάση n, και ένα έγκυρο ακέραιο θετικό αριθμό στη βάση n από τη γραμμή εντολής και εκτυπώνει το δεκαδικό ισοδύναμο του συγκεκριμένου αριθμού. Πιο κάτω δίνονται ορισμένα παραδείγματα χρήσης του προγράμματος αυτού:

```
$ java BaseConvert 2 1101101
Number 1101101 in base 2 equals 109 in base 10

$ java BaseConvert 1 0
Wrong base 1

$ java BaseConvert 5 345
Wrong number 345 in base 5

$ java BaseConvert 5 12334
Number 12334 in base 5 equals 969 in base 10
```

Για την απάντησή σας συμπληρώστε το ακόλουθο πρόγραμμα:

```
public class BaseConvert{

 public boolean valid_num(int base, int num){

/* Η συνάρτηση valid_num λαμβάνει μια έγκυρη βάση base
   ανάμεσα στο 2 και το 9 και ένα αριθμό num και
   επιστρέφει true αν ο αριθμός num είναι έγκυρα
   διατυπωμένος στη βάση base, δηλαδή όλα τα ψηφία του
   είναι από το πεδίο [0 .. base-1]. Διαφορετικά η
   συνάρτηση επιστρέφει false. Για παράδειγμα, ο
   αριθμός 345 δεν είναι έγκυρα διατυπωμένος στη βάση 5
```

```
   αφού περιλαμβάνει το ψηφίο 5, ενώ ο αριθμός 12334
   είναι έγκυρα διατυπωμένος στη βάση 5.
*/
                    . . . . . .

}

public static int base_n_to_decimal (int base, int num){

/* Η συνάρτηση base_n_to_decimal λαμβάνει μια έγκυρη βάση
   base ανάμεσα στο 2 και το 9 και ένα θετικό ακέραιο
   αριθμό, num,έγκυρα διατυπωμένο στη βάση base, και
   επιστρέφει τον ισοδύναμο δεκαδικό αριθμό του num.
*/

                    . . . . . .
}

public static void main (String[] args) {

/* Η συνάρτηση main λαμβάνει από τη γραμμή εντολής τη
   βάση που πρέπει να είναι ανάμεσα στο 2 και το 9, και
   ένα ακέραιο θετικό αριθμό στην εν λόγω βάση, και
   νοουμένου ότι τα στοιχεία αυτά είναι έγκυρα, προχωρεί
   στη μετατροπή του αριθμού σε δεκαδική βάση.
*/

 int base = Integer.parseInt(args[0]);
 int num = Integer.parseInt(args[1]);

                    . . . . . .
 }
}
```

Ερώτηση 4

Κατασκευάστε ένα αρθρωτό πρόγραμμα σε Java (Draw.java) το οποίο δημιουργεί διακριτά σχήματα σύμφωνα με τις επιλογές του χρήστη. Συγκεκριμένα υπάρχουν τρεις έγκυρες επιλογές, οι οποίες αντιπροσωπεύονται από τους αριθμούς 1, 2 και 3 και εξηγούνται στη συνέχεια.

Η επιλογή νούμερο 1 δημιουργεί διακριτά τετράγωνα συγκεκριμένης μορφής και μεγέθους. Τόσο η επιλογή (1), όσο και το μέγεθος (ένας φυσικός αριθμός) δίνονται από το χρήστη σε αυτή τη σειρά, στη γραμμή εντολής, όπως επιδεικνύεται πιο κάτω:

```
$ java Draw 1 3
```

```
++/
+/-
/--

$ java Draw 1 5

++++/
+++/-
++/--
+/---
/----
```

Η επιλογή νούμερο 2 δημιουργεί σειρές δεδομένων μεγεθών, βάσει τριών υπο-επιλογών που αντιπροσωπεύονται από τους αριθμούς 1, 2, και 3, όπως επιδεικνύεται πιο κάτω:

```
$ java Draw 2 1 11 7 5 3 1 3 5 7 11

***********
*******
*****
***
*
***
*****
*******
***********

$ java Draw 2 2 11 7 5 3 1 3 5 7 11

***********
******************
************************
**************************
****************************
********************************
************************************
****************************************
********************************************************

$ java Draw 2 3 11 7 5 3 1 3 5 7 11

*
***
***
*****
*****
*******
*******
***********
***********
```

Επισημαίνεται ότι όλα τα στοιχεία δίνονται στη γραμμή εντολής. Ο πρώτος αριθμός, όπως και προηγουμένως είναι η συγκεκριμένη επιλογή (2). Ο δεύτερος αριθμός είναι η σχετική υπο-επιλογή (1 στο πρώτο παράδειγμα, 2 στο δεύτερο παράδειγμα και 3 στο τρίτο παράδειγμα). Οι υπόλοιποι αριθμοί (οποιουδήποτε πλήθους) αντιπροσωπεύουν τα μήκη των σειρών που χρειάζεται να δημιουργηθούν. Για σκοπούς πιο άμεσης σύγκρισης στα τρία παραδείγματα αυτής της επιλογής έχουν χρησιμοποιηθεί οι ίδιες σειρές, συγκεκριμένα οι σειρές με μήκη 11, 7, 5, 3, 1, 3, 5, 7, 11. Στην πρώτη υπο-επιλογή (1), απλά παρουσιάζονται οι σειρές όπως δίνονται. Στη δεύτερη υπο-επιλογή (2) τα μήκη παρουσιάζονται συσσωρευτικά από το πρώτο μέχρι το τελευταίο. Επομένως στο παράδειγμα δημιουργούνται σειρές με μήκη 11, 18, 23, 26, 27, 30, 35, 42 και 55. Τέλος στην τρίτη υπο-επιλογή (3) τα δεδομένα μήκη ταξινομούνται από το μικρότερο στο μεγαλύτερο και με αυτή την κατάταξη δημιουργούνται οι σειρές, δηλαδή με μήκη 1, 3, 3, 5, 5, 7, 7, 11. Για την απαιτούμενη ταξινόμηση δεν χρειάζεται να ορίσετε συνάρτηση αλλά να κάνετε χρήση της ακόλουθης συνάρτησης sort, της οποίας όμως χρειάζεται να εξηγήσετε συνοπτικά τη λειτουργία:

```
public static void sort(int[] T){
   int L = T.length; boolean sorted;
   do{
       sorted = true;
       for (int i = 0; i < L-1; i++){
           if (T[i] > T[i+1]){
              sorted = false;
              int temp = T[i]; T[i] = T[i+1];
                  T[i+1]=temp;
           }
       }
       L = L-1;
   } while(!sorted && L > 1);
}
```

Η επιλογή νούμερο 3 της Draw δημιουργεί μια σειρά από σταυρούς δεδομένου μεγέθους και δεδομένου πλήθους όπως επιδεικνύεται στα ακόλουθα παραδείγματα:

```
$ java Draw 3 3 1

    |
    |
    |
---+---
    |
    |
    |
```

```
$ java Draw 3 1 10

 |   |   |   |   |   |   |   |   |   |
-+- -+- -+- -+- -+- -+- -+- -+- -+- -+-
 |   |   |   |   |   |   |   |   |   |

$ java Draw 3 4 2

     |          |
     |          |
     |          |
     |          |
----+---- ----+----
     |          |
     |          |
     |          |
     |          |
```

Επισημαίνεται ότι το μέγεθος του σταυρού δίνεται ως το δεύτερο όρισμα στη γραμμή εντολής και το πλήθος των σταυρών ως το τρίτο όρισμα στη γραμμή εντολής. Συνεπώς στο πρώτο παράδειγμα αυτής της επιλογής της Draw δημιουργείται μόνο ένας σταυρός μεγέθους 3. Στο δεύτερο παράδειγμα δημιουργείται μια σειρά από δέκα (10) σταυρούς μεγέθους 1, και στο τρίτο παράδειγμα δημιουργείται μια σειρά από δύο (2) σταυρούς μεγέθους 4.

Για την κατασκευή των σειρών όλων των πιο πάνω σχημάτων, για όλες δηλαδή τις επιλογές/υπο-επιλογές της Draw, μπορούν να αξιοποιηθούν οι ακόλουθες δύο συναρτήσεις:

```
public static void putchar (char c){System.out.print(c);}

public static void Line (int len, char c){
   for (int i = 1; i <= len; i++) putchar(c);
}
```

ΤΕΛΟΣ ΕΡΩΤΗΣΕΩΝ

ΧΩΡΟΣ ΑΠΑΝΤΗΣΕΩΝ

Ονοματεπώνυμο Φοιτητή: ----------

Ταυτότητα: ----------

Υπογραφή: ----------

Ερωτήσεις	**Μονάδες**
Ερώτηση 1	
Ερώτηση 2	
Ερώτηση 3	
Ερώτηση 4	
Σύνολο Μονάδων	

Παρατηρήσεις Διδάσκοντα

Απάντηση στην Ερώτηση 1

(τυχόν συνέχεια στην απάντηση της Ερώτησης 1)

Απάντηση στην Ερώτηση 2

Μέρος (α)

Μέρος (β)

(τυχόν συνέχεια στην απάντηση της Ερώτησης 2)

Απάντηση στην Ερώτηση 3

(τυχόν συνέχεια στην απάντηση της Ερώτησης 3)

Απάντηση στην Ερώτηση 4

(τυχόν συνέχεια στην απάντηση της Ερώτησης 4)

Επιπρόσθετο φύλλο (για συμπλήρωση απάντησης ή πρόχειρο)

Επιπρόσθετο φύλλο (για συμπλήρωση απάντησης ή πρόχειρο)

Επιπρόσθετο φύλλο (για συμπλήρωση απάντησης ή πρόχειρο)

Επιπρόσθετο φύλλο (για συμπλήρωση απάντησης ή πρόχειρο)

ΤΕΛΟΣ ΕΞΕΤΑΣΤΙΚΟΥ ΔΟΚΙΜΙΟΥ