

ESC101 : Fundamental of Computing

End Semester Exam

19 November 2008

Solution and Grading Policy

Name :

Roll No. :

Section :

Note : Read the instructions carefully

1. You will lose 3 marks if you forget to write your name, roll number, and section.
2. There are **EIGHT** questions in this paper distributed among total five sheets of the answer sheet.
3. You have to write in the space provided. Please avoid over writing.
4. You will be given rough sheets.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Marks								

1. (marks = 5)

What is the value of `d` at the end of each of the following code fragments ? If you feel that there will be any compilation error, then mention that the cause of compilation error.

(a) `double d;`

`d = 2+11/9*4.5;`

(b) `int i=10; byte b=12;`

`float d = b+i+1.3;`

(c) `long d;`

`d = 1234567898765;`

(d) `double A=11; byte b=2;`

`float d = (float)(A/b/2*4);`

(e) `int d;`

`d = (int)3.4/1.1;`

Answer :

(a) **6.5**

(b) **Compilation error :** Expression on right hand side is of type `double` whereas variable on the left hand side is of type `float`. It is not permitted to assign `double` to `float` without explicit type cast

(c) **Compilation error :** The literal on the right hand side is beyond the range of `int`. There should be `L` at the end of literal.

(d) **11.0**

(e) **Compilation error :**

`(int)3.4/1.1` is equal to `3/1.1` since type cast operator has higher precedence than `/` operator. Since `3/1.1` is of type `double`, so it cannot be assigned to `d` which is of type `int`.

Grading Policy : Each blank has one mark. In case of compilation error, 0.5 marks are for mentioning the cause/nature of the compilation error. You should mention it CLEARLY.

2. (marks = 8)

Given two arrays A and B of size m and n respectively and $m \geq n$. The arrays are storing characters. We say that B appears in A if there exists a non-negative integer i such that for all j with $0 \leq j < n$, $A[i + j] = B[j]$. For example if $A=\{s, t, a, x, n, b, y\}$, and $B=\{a, x, n\}$, then B appears in A. Write a method `Appears` which takes two character arrays A and B as parameters and returns true if B appears in A and false otherwise. You may assume that $A.length \geq B.length$ is ensured by the method which calls the method `Appears`.

Answer :

```
public static boolean Appears(char[] A, char[] B)
{
    int m = A.length;
    int n = B.length;
    boolean appears = false;

    for(int i=0;i<=m-n && appears==false; i=i+1)
    {
        int j = 0;
        while(j<n && A[i+j]==B[j])
            j=j+1;
        if(j==n) appears = true;
    }
    return appears;
}
```

Grading Policy : This question tests whether you can write programs for reasonably simple problems. It tests your algorithmic skill and knowledge of JAVA.

There may be multiple correct solutions. **2 marks** are deducted if the terminating condition of the outer `for` loop is incorrect. More specifically, $i \leq m - n$ must be one of the terminating conditions. It is wrong if one writes $i < m - n$ or $i < m$.

For other logical or syntax errors too, there is suitable penalty.

3. (marks = 8)

The following method computes product of two matrices M and N and returns the reference to the final product matrix. Fill in the blanks appropriately.

```
// this method assumes that M and N are matrices
// and that columns of M = rows of N.
// It returns reference to the matrix which corresponds to M*N

public static int[ ][ ] multiply(int[ ][ ] M, int[ ][ ] N)
{
    int rows = M.length;

    int columns = N[0].length;

    int[ ][ ] R = new int[rows][columns];

    for(int r=0; r < rows ; r = r+1)
    {
        for(int c=0; c < columns ; c=c+1)
        {
            int product = 0;

            for(int i=0; i < N.length ;i=i+1)
            {
                product = product + M[r][i]*N[i][c];
            }

            R[r][c]= product;
        }
    }
    return R;
}
```

Grading Policy : The first blank and the last three blanks carry 0.5 marks each. Every other blank carried one mark. Note that it is incorrect to use `N[1].length` instead of `N[0].length` since the matrix N could have just a single column.

4. (marks=6)

Recall the class *Triangle* which was described in details during some lecture. For your assistance, the following was the skeleton of the *Triangle* class.

```
class Triangle
{
    private Point P;
    private Point Q;
    Private Point R;

    public Triangle()
    {
        .....
    }

    public Triangle(Point A, Point B, Point C)
    {
        .....
    }

    public double Area()
    {
        .....
    }

    public double Perimeter()
    {
        .....
    }
}
```

We want to add one more non-static method `Encloses()` to this class. This method takes a point as parameter and returns true if the point is enclosed by the triangle and false otherwise. Note that a point lying on one of the sides of the triangle is also said to be enclosed by the triangle. Please fill in the blanks of the following skeleton of the method. **You may write at most one statement/expression in a blank.** You can't use `&&` or `||` in the expression of if statement.

```
public boolean Encloses(Point O)
{

    Triangle one = new Triangle(this.P, this.Q, O);

    Triangle two = new Triangle(this.P, this.R, O);

    Triangle three = new Triangle(this.Q, this.R, O);

    if(this.Area() == one.Area() + two.Area() + three.Area())

        return true;

    else return false;
}
```

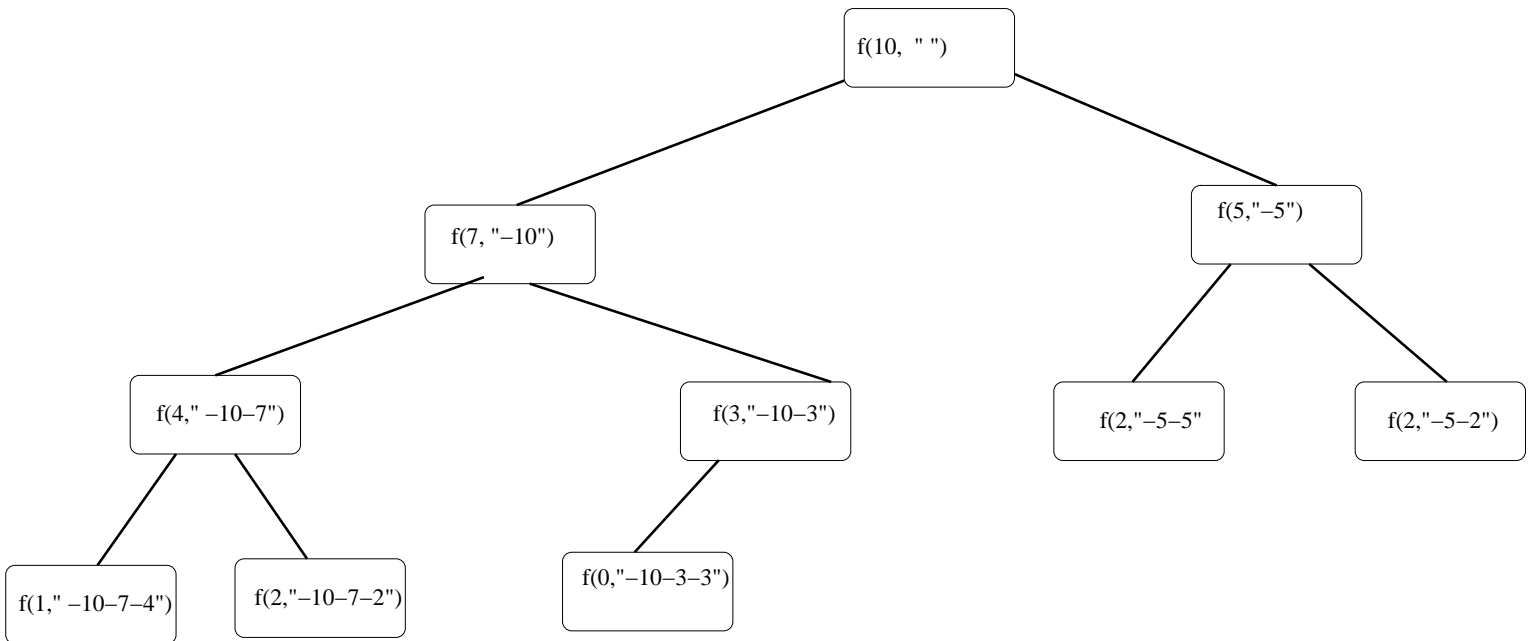
Grading Policy : 1 mark for first three blanks and 3 marks for the blank in the condition of if statement. Also note that it is fine if one writes `P` instead of `this.P` (and similarly for `Q`, and `R`). If some one is computing slope of line segments in his/her solution, the resulting solution is not correct because it will fail if one of the line segment of the triangle is vertical. **There is no partial credits for such a incomplete method which works only for some inputs.**

5. (Marks = 3,5)

Draw the recursion tree for the following method when invoked with arguments $n = 10$, and $S = ""$. Also print the output of the method.

```
public static void f(int n, String S)
{
    if(n<=2) System.out.println(S);
    else
    {
        f(n-3, S+"-"+n);

        if(n>=4) f(n/2,S+"-"+(n/2));
    }
}
```



Output :

-10-7-4

-10-7-2

-10-3-3

-5-5

-5-2

Grading Policy :

3 marks for the recursion tree. Five marks for the output printed. 2 marks are deducted if the output is reversed.

6. (7 marks)

There is a function G defined for positive integers as follows.

$$G(n) = \begin{cases} 4 & \text{if } 1 \leq n \leq 4 \\ G(n-1) + G(n-2) + 2G(n-3) - 2 & \text{if } n > 4 \end{cases}$$

Write a method for computing $G(n)$ which performs at most $c n$ instructions for some constant c .

```
public static int G(int n)
{
    if(n<=4) return 4;

    else
    {
        int a = 4;

        int b = 4;

        int c = 4;

        for(int i=5 ; i<=n ; i = i+1)
        {
            int temp = c+b+2*a-2;

            a = b;

            b = c;

            c = temp;
        }

        return c;
    }
}
```

Grading Policy :

Iterative solution is the correct solution. The recursive solution will have exponential (in terms of n) running time. The argument for this is similar to that of recursive solution of **fibonacci** number. Recall that the complete analysis of the latter problem was provided in the solution to the practise problems.

0 marks for recursive solution.

7. (Marks = 1,3,3,1)

Given a grid with height n and width $2n$, We want to enumerate all those shortest paths from origin $(0,0)$ to $(2n,n)$ which satisfy the constraint :

Each point of the path is either on or below the diagonal.

See figure given below for better understanding. The program enumerates a path so that it prints `-up` for a step in upward direction and prints `-right` for a step in right direction. For example for $n = 2$, the output should be

```
-right-right-up-right-right-up
-right-right-right-up-right-up
-right-right-right-right-up-up
```

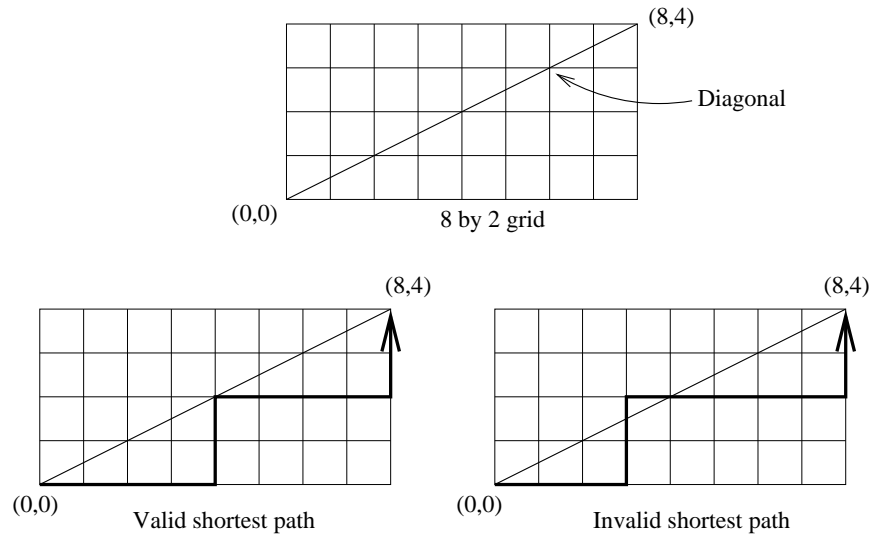


Figure 1: The grid, an example of valid path and an example of invalid path

Please fill in the blanks of the following program carefully. The program reads the value of n from command line.

```
class special_paths
{
    public static void PathS(int r, int u, String S)
    {
        if(r==0 && u==0) System.out.println(S);

        else
        {
            if(2*(u-1)>=r) PathS(r,u-1,S+"-up");

            if(r>0) PathS(r-1,u,S+"-right");

        }
    }

    public static void main(String args[])
    {
        int n = Integer.parseInt(args[0]);
        if(n<1) System.out.println("Invalid input");
        else
            PathS(2*n, n, " ");
    }
}
```

Grading Policy : marks distributed according to the order of blanks. Note that the order of the two statements in the `else` part does not matter.

8. (10) The idea underlying binary search is a very useful idea and can be used for a variety of problems. We showed that we can use it for the following problem. Given an array A which stores 0's and 1's such that all 0's precede all 1's, find the smallest index of the array which stores 1. We are going to use similar idea for the following problem.

There is an array A storing n distinct integers. The numbers are arranged such that there is a non-negative integer $p < n$ such that $A[0], \dots, A[p]$ is a strictly increasing sequence and $A[p], \dots, A[n-1]$ is a strictly decreasing sequence of numbers. The following code prints the value of p . Fill in the blanks accordingly.

```
public static void Largest_value(int[] A)
{   int n = A.length;
    int mid = 0;

    if(A[0]>A[1]) System.out.println("The value of p = 0");

    else
    {   if(A[n-1]>A[n-2]) System.out.println("The value of p =''+(n-1));

        else
        {   boolean p_is_found = false;

            int left = 1 ;

            int right = n-2 ;

            while(p_is_found == false)
            {

                mid = (left+right)/2;

                if(A[mid]>A[mid-1])
                {

                    if(A[mid]>A[mid+1]) p_is_found = true;

                    else    left = mid + 1;
                }

                else right = mid-1;
            }

            System.out.println("The value of p is "+mid);

        }
    }
}
```

Grading policy : Each blank carries **one** mark. There may be more than one correct solution.

For the first blank, it is fine if some one has written `n==1 || A[0]>A[1]` instead of just `A[0]>A[1]`.

Note : The value of `left` must not be initialized to 0. To convince yourself, try executing the above code on `{3, 10, 9, 6, 4, 2}` with `left` initialized to 0. It will raise `ArrayIndexOutOfBoundsException`.