ESc 101: ALGORITHMS AND PROGRAMMING

Course Instructor : Prof. Y. Narahari

Due Date : 2 September, 2011

INSTRUCTIONS: Please attempt these questions individually. Group Discussion is **NOT** encouraged. You can submit this anytime before the due date and get a feedback on your answers. This exercise would be a stepping stone for future work.

1. Time for some design! It's your lucky day today! You are gifted some stars today and you are told to place them in the Outer Space in the following way:

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

We can help you with the algorithm, but the main job of coding is yours! As we can see, there are four spaces to the left of the topmost star. Also, there are three spaces to the left of the first star in the next row and so on. So, there are (5-i) spaces to the left of the first star in the 'i'th row. You may use variable 'j' for putting spaces to the left of the first stars in each row. We can also see that the number of stars increases by one as we go down the row. So, once we are in the 'i'th row, after giving the required number of spaces to the left, we can now place (i) number of '* ' (star followed by a space). You may use variable 'k' for placing the '* '. After you have placed the required number of stars in that row, go to the next row. Tell the computer to keep this going on for five rows and.. That's it!

Here is your pseudo-code for 5 rows of stars:

```
for i = 1 to 5
{
  for j = 1 to (5-i)
  {
    give space
  }
  for k = 1 to i
  {
    place '* ' (star followed by a space)
  }
  go to the next line
}
```

Write a C program which accepts n from user and displays n rows of star.

2. **Algorithm for generating the first n Fibonacci numbers**

Theory: In mathematics, the Fibonacci numbers are the numbers in the following integer sequence:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

By definition, the first two Fibonacci numbers are 0 and 1, and each subsequent number is the sum of the previous two.

In mathematical terms, the sequence Fn of Fibonacci numbers is defined by the recurrence relation

$$F_n = F_{n-1} + F_{n-2} \tag{1}$$

with seed values $F_0 = 0$ and $F_1 = 1$.

Algorithm (With explanation): From the definition we are given that: New term = "preceding term" + "term before preceding term". Assume
a : As the term before preceding term
b : As the preceding term
c : New term

To start with we have:

i) a := 0 first Fibonacci number
ii) b := 1 second Fibonacci number
iii) c := a+b third Fibonacci number (from definition)

The fourth Fibonacci is derived from the sum of the second and third Fibonacci numbers.With regard to the definition, the second Fibonacci number has the role of term before the preceding term and the 3rd Fibonacci number has the role preceding term therefore, before making the next computation we must ensure that:

iv) a := b i.e. ”term before preceding term” becomes preceding term
v) b := c i.e ”preceding term” becomes new term.

To generate next number, loop back to step iii. The complete algorithm is:

```
a := 0;
b := 1;
i :=  2;
while i<n
        begin
            i := i+1;
            c := a+b;
            display c;
            a := b;
            b := c
        end
```

3. What does the following program do? Please do **NOT** copy and run it, try to trace it out yourself.

```
#include<stdio.h>

main()
{
  int a[101][101],b[101][101],i,j,m,n;

  for(i=0;i<101;i++)
  {
    for(j=0;j<=i;j++)
    {

      a[i][j] = (i*i*i)+(j*j*j);
      b[i][j] = a[i][j];

    }
  }

  printf("\n");

  for(i=0;i<101;i++)
  {
    for(j=0;j<=i;j++)
    {
      for(m=0;m<101;m++)
      {
        for(n=0;n<=m;n++)
        {
          if((a[i][j]==b[m][n]) && (i!=m) && (j!=n) && (a[i][j]<=1000000)
                                                      && (b[m][n]!=0))
            printf("%d\n",a[i][j]);
          b[i][j]=0;
        }
      }
    }
  }

}
```

4. What is the output of the program? As before do **NOT** run it. What do you think is happening here?

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

```
main()
{
  int row, col, array[5][5], cycle=1, deadend=0;

  for(row=0; row<5; row++)
  {
    for(col=0; col<5; col++)
    {
      array[row][col] = row*5 + col + 1;
    }
  }

  row=0;
  col=0;

  while(cycle<10)
  {
    printf("%d ", array[row][col]);
    if(cycle%2 == 0)
    {
      col--;
      row++;
    }
    else
    {
      row--;
      col++;
    }

    if(row>=5)
    {
      row=4;
      col+=2;
      deadend=1;
    }
    else if(col>=5)
    {
      col=4;
      row+=2;
      deadend=1;
    }
    else if(row<0)
    {
      row=0;
      deadend=1;
    }
    else if(col<0)
    {
      col=0;
      deadend=1;
    }
    else
    {
      /* No dead end. Go ahead! */
    }
    if(deadend == 1)
    {
      cycle++;
      deadend=0;
    }
  }
  printf("\n");
}
```

5. Reading Assignment 2.1, 2.2, 2.4, 2.5, 2.11 & chapter 3. Please use the lab time to ask the TAs any question you may have on these chapters.