Software Engineering

-supervised by:

Dr. Suparna Biswas (Saha)

Rohit Das

B. Tech(Computer Sc. and Engg)

Roll: 30000114022

Regn. No.:143000110023

6th Semester,2016

Maulana Abul Kalam Azad University of Technology,

West Bengal.

LaTeX 2017

Contents

1. Assignment-1: C programming (Date: 27/10/2017)

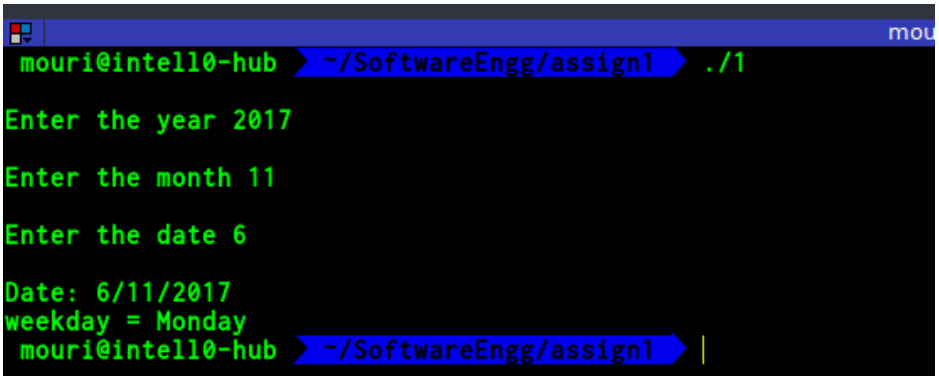
1..1 Write a program to implement Calendar program to display Day of the month. Program will accept year, month and date from user and will display the day of the month.

Program:

```
/*
* Copyright Rohit Das (C) 2017
* To display day from date
*/
#include<stdio.h>
#include<math.h>

int fm(int date, int month, int year) {
   int _month, leap;
   //leap function 1 for leap & 0 for non-leap
   if ((year % 100 == 0) && (year % 400 != 0))    leap = 0;
   else if (year % 4 == 0)    leap = 1;
   else    leap = 0;
   _month = 3 + (2 - leap) * ((month + 2) / (2 * month))
         + (5 * month + month / 9) / 2;
   _month %= 7;
   return _month;
}

int day_of_week(int date, int month, int year) {
   int day;
   int yy = year % 100;
   int century = year / 100;
   printf("\nDate: %d/%d/%d \n", date, month, year);
   day = 1.25 * yy + fm(date, month, year) + date - 2 * (century % 4);
   //remainder on division by 7
   day %= 7;
   switch (day) {
      case 0:
         printf("weekday = Saturday");    break;
      case 1:
         printf("weekday = Sunday");    break;
      case 2:
         printf("weekday = Monday");    break;
      case 3:
         printf("weekday = Tuesday");    break;
      case 4:
         printf("weekday = Wednesday");    break;
      case 5:
         printf("weekday = Thursday");    break;
      case 6:
         printf("weekday = Friday");    break;
      default:
         printf("Incorrect data");
   }
```

```
    printf("\n");    return 0;
}

int main() {
    int date, month, year;
    printf("\nEnter the year: ");    scanf("%d", &year);
    printf("\nEnter the month: ");    scanf("%d", &month);
    printf("\nEnter the date: ");    scanf("%d", &date);
    day_of_week(date, month, year);
    return 0;
}
```

Output:



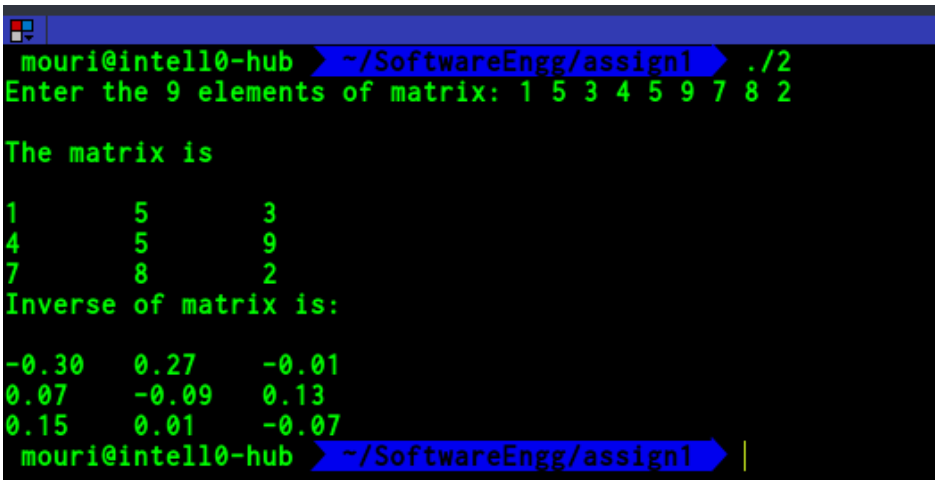
1..2 Write a program to find inverse of 3x3 matrix.

Program:

```
/*
* Copyright Rohit Das (C) 2017
* To display inverse of a 3x3 matrix
*/
#include<stdio.h>

int main(){
  int a[3][3],i,j;
  float determ=0;
  printf("Enter the 9 elements of matrix: \n");
  for(i=0;i<3;i++)      for(j=0;j<3;j++)      scanf("%d",&a[i][j]);
  printf("\nThe matrix is:\n");
  for(i=0;i<3;i++){
      printf("\n");
      for(j=0;j<3;j++)      printf("%d\t",a[i][j]);
  }
  for(i=0;i<3;i++)     determ += (a[0][i]*(a[1][(i+1)%3]*a[2][(i+2)%3] -
  a[1][(i+2)%3]*a[2][(i+1)%3]));
   printf("\nInverse of matrix is: \n\n");
   for(i=0;i<3;i++){
      for(j=0;j<3;j++)
      printf("%.2f\t",((a[(i+1)%3][(j+1)%3] * a[(i+2)%3][(j+2)%3]) -
      (a[(i+1)%3][(j+2)%3]*a[(i+2)%3][(j+1)%3]))/ determ);
```

```
        printf("\n");
    }
    return 0;
}
```

Output:



1..3 Write a program to check whether matrix is Magic Square or not.

Program:

```
/*
* Copyright Rohit Das (C) 2017
* To check if a matrix is Magic Square
*/
#include<stdio.h>
#include<stdlib.h>

void input();   void init();   void getdata();   void validate();
void validmat();   int leftdia();   int rightdia();   int rowsum();
int colsum();

//global vars
int arr[100][100],size;

//functions
void input(){
        int num=0;
        printf("Enter magic square size: \n");  scanf("%d",&size);
        init(size,num);
        printf("enter the elements: \n");       getdata(size);
}

void init(int size,int num)
{
        int i,j;
        for(i=0;i<size;i++)     for(j=0;j<size;j++)     arr[i][j]=num;
}
```

```
void getdata(int size)
{
        int i,j,num;
        for(i=0;i<size;i++){
                for(j=0;j<size;j++){
                        scanf("%d",&num);       arr[i][j]=num;
                }
        }
}

void validate()
{
    int i,j,k;     int ar[size*2+2];
    int arsize=(size*2+2);
    for(i=0;i<(size*2+2);i++)     ar[i]=0;
    for(i=0;i<size;i++)     ar[i]=rowsum(arr,i);
    for(j=0;j<size;j++)     ar[j+i]=colsum(arr,j);
    ar[j+i]=leftdia();     ar[j+i+1]=rightdia();
        for (i = 0;i < size; i++) {
                for (j = 0;j < size; j++) printf("%d ",arr[i][j]);
                printf("\n");
        }
    validmat(ar,arsize);
}

void validmat(int ar[],int arsize)//validation
{
    int i,valid=0;
    for(i=0;i<arsize-1;i++)
    {
        if(ar[i]==ar[i+1])     valid=1;
        else     valid=0;
    }
    if(valid==1)     printf("This matrix is a magic square.\n");
    else    printf("This matrix is not a magic square.\n");
}

int leftdia()    //left diagonal sum
{
    int i,sum=0;
    for(i=0;i<size;i++)     sum=sum+arr[i][i];
    return sum;
}

int rightdia()  //right diagonal sum
{
    int i,sum=0;
    for(i=0;i<size;i++)     sum=sum+arr[i][size-1-i];
    return sum;
}
```

```c
int rowsum(int arr[100][100],int index) //calculates row sum
{
    int i,x=0;
    for(i=0;i<size;i++)     x=x+arr[index][i];
    return x;
}

int colsum(int arr[100][100],int index)//calculates sum of the columns
{
    int i,x=0;
    for(i=0;i<size;i++)     x=x+arr[i][index];
    return x;
}

int main()
{
        input();     validate();
        return 0;
}
```

Output:

1..4 Write a program to read last n characters from a file(input should be a.txt).

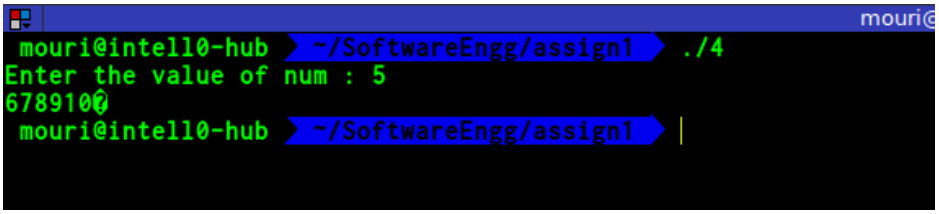
Program:

```c
/*
* Copyright Rohit Das (C) 2017
* a.txt for 4.c
*/
12345678910
```

```c
/*
* Copyright Rohit Das (C) 2017
* To read last n characters from a file (used a.txt)
*/
#include <stdio.h>
#include <stdlib.h>

int main() {
```

```
    FILE *fp;    char ch;
    int num;    long length;
    printf("Enter the value of num : ");
    scanf("%d", &num);
    fp = fopen("a.txt", "r");              //open the file
    if (fp == NULL) {
        puts("cannot open this file");    exit(1);
    }
    fseek(fp, 0, SEEK_END);          //set fp to end of file
    length = ftell(fp);
    fseek(fp, (length - num - 1), SEEK_SET);//2 because of EOF
    do {
        ch = fgetc(fp);    putchar(ch);
    } while (ch != EOF);
    printf("\n");
    fclose(fp);    return(0);
}
```

Output:



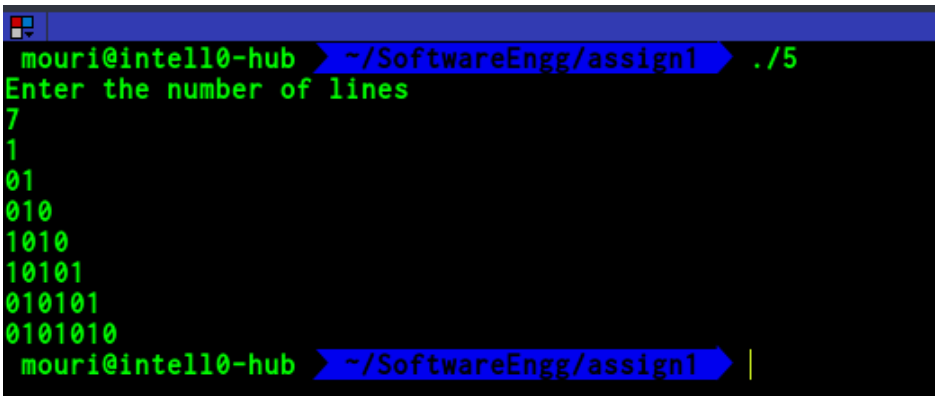
1..5 Write a program to print binary numbers in pyramid pattern.

Program:

```
/*
* Copyright Rohit Das (C) 2017
* To display binary pattern as shown:
1
0 1
1 0
1 0 1
0 1 0 1
1 0 1 0 1
*/
#include<stdio.h>
#include<stdlib.h>

int main()
{
        int n,i,j,k;    k = 1;
        printf("Enter the number of lines\n");    scanf("%d",&n);
        for(i=0;i<n;i++) {
        for(j=0;j<=i;j++) {
                printf("%d",k);
                if(k==1)    k=0;
```

```
                else    k=1;
            }
        printf("\n");
        }
        return 0;
}
```

Output:



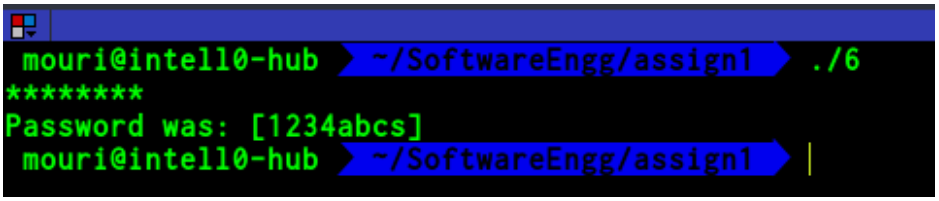
1..6 Write a program to input password for validation of username.
Enter password: ****
Password entered: sourav

Program:

```
/*
* Copyright Rohit Das (C) 2017
* To validate entered password
*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <termios.h>
#include <ctype.h>


/* Use this variable to remember original terminal attributes. */
struct termios saved_attributes;

void reset_input_mode (void)
{
  tcsetattr (STDIN_FILENO, TCSANOW, &saved_attributes);
}

void set_input_mode (void)
{
        struct termios tattr;  char *name;
        /* Make sure stdin is a terminal. */
        if (!isatty (STDIN_FILENO)){
              fprintf (stderr, "Not a terminal.\n");
```

```
      exit (EXIT_FAILURE);
        }
        /* Save the terminal attributes so we can restore them later. */
        tcgetattr (STDIN_FILENO, &saved_attributes);
        atexit (reset_input_mode);
        /* Set the funny terminal modes. */
        tcgetattr (STDIN_FILENO, &tattr);
        tattr.c_lflag &= ~(ICANON | ECHO); /* Clear ICANON and ECHO. */
        tattr.c_cc[VMIN] = 1;
        tattr.c_cc[VTIME] = 0;
        tcsetattr (STDIN_FILENO, TCSAFLUSH, &tattr);
}

int main ()
{
        int i = 0;
        char c, password[100], asterisk = '*';
        set_input_mode ();
        while (read (STDIN_FILENO, &c, 1) && (isalnum (c) || ispunct (c))
         && i < sizeof (password) - 2)
    {
      password[i++] = c;
      write (STDOUT_FILENO, &asterisk, 1);
    }
        password[i] = 0;
        printf ("\nPassword was: [%s]\n", password);
        return EXIT_SUCCESS;
}
```

Output:



1..7 Write a program to create your own header file in C.

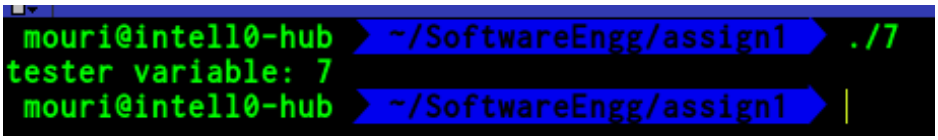
Program:

```
/*
* Copyright Rohit Das (C) 2017
* test.h for 7.c
*/
#ifndef TEST_H
#define TEST_H

int tester;

void test_it_out();
```

```
#endif
```

```
/*
* Copyright Rohit Das (C) 2017
* To create a header file and use it (used test.h)
*/
#include "test.h"
#include <stdio.h>

void test_it_out()
{
        tester = 7;     printf("tester variable: %d\n", tester);
}

int main()
{
        test_it_out();
        return 0;
}
```

Output:



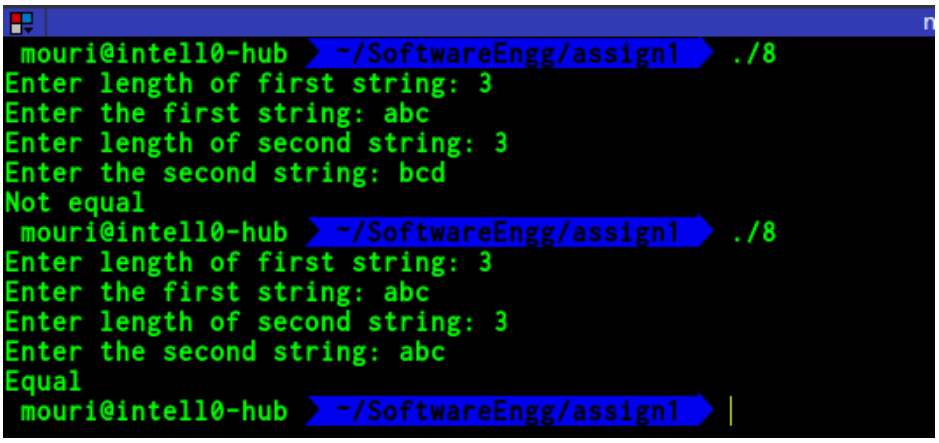
1..8 Write a program to compare two strings without using library function(strcmp).

Program:

```
/*
* Copyright Rohit Das (C) 2017
* To compare two strings without strcmp
*/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int check(char* s1, char* s2, int len1, int len2)
{
        if (len1 != len2) {
                printf("Not equal\n");
                return 1;
        }

        for (int i = 0; i < len1; i++) {
                if (s1[i] != s2[i]) {
                        printf("Not equal\n");
                        return 1;
                }
        }
        printf("Equal\n");
```

```
        return 0;
}

int main()
{
        int len1, len2;
        printf("Enter length of first string: ");
        scanf("%d", &len1);
        printf("Enter the first string: ");
        char* s1 = (char *)malloc((len1 + 1) * sizeof(char));
        scanf("%s", s1);
        printf("Enter length of second string: ");
        scanf("%d", &len2);
        printf("Enter the second string: ");
        char* s2 = (char *)malloc((len2 + 1) * sizeof(char));
        scanf("%s", s2);

        check(s1, s2, len1, len2);
        return 0;
}
```

Output:



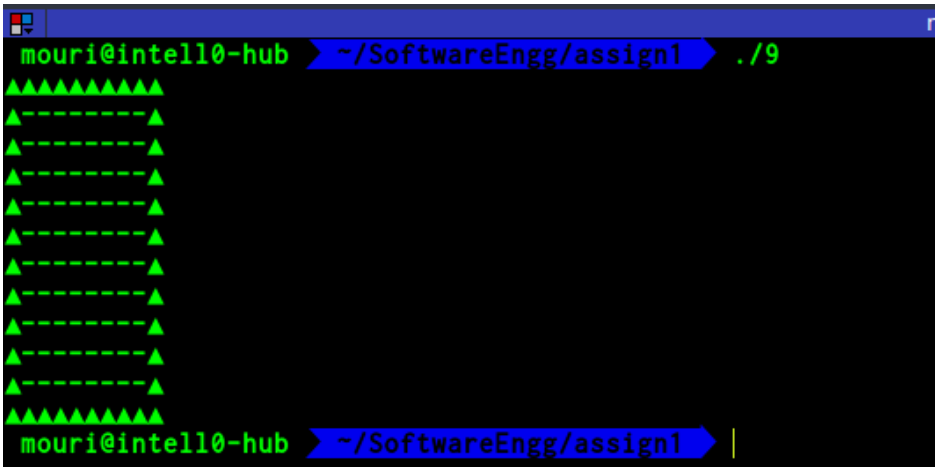
1..9 Write a program to print a rectangle using line and special symbols.

Program:

```
/*
* Copyright Rohit Das (C) 2017
* To display pattern as shown below

__________
__________
__________
__________
__________
__________
__________
__________
```

```c
*/
#include <locale.h>
#include <stdio.h>
#include <wchar.h>

#define DIM 10

int main(void)
{
    setlocale(LC_ALL, "");
    for (int i = 0; i < DIM; i++) {
        printf("%lc", (wint_t)9650);
    }
    printf("\n");

    for (int j = 0; j < DIM; j++) {
            for (int i = 0; i < DIM; i++) {
                if (i == 0 || i == DIM -1) {
                        printf("%lc", (wint_t)9650);
                        if (i == DIM - 1) {
                                printf("\n");
                        }
                } else {
                        printf("-");
                }
            }
    }

    for (int i = 0; i < DIM; i++) {
        printf("%lc", (wint_t)9650);
    }
    printf("\n");
    return 0;
}
```

Output:



1..10 Write a program to find addition of lower triangular matrix.

Program:

```
/*
* Copyright Rohit Das (C) 2017
* To display addition of lower triangular elements
*/
#include <stdio.h>
#include <stdlib.h>

int main()
{
        int i, j;

        int size;
        printf("Enter size: ");
        scanf("%d", &size);

        int **a = (int **) malloc(size * sizeof(int *));
        for (i = 0; i < size; i++) {
                a[i] = (int *)malloc(size * sizeof(int));
        }

        int **b = (int **) malloc(size * sizeof(int *));
        for (i = 0; i < size; i++) {
                b[i] = (int *)malloc(size * sizeof(int));
        }

        int **c = (int **) malloc(size * sizeof(int *));
        for (i = 0; i < size; i++) {
                c[i] = (int *)malloc(size * sizeof(int));
        }

        printf("Enter the elements for first matrix:\n");
        for (i = 0; i < size; i++) {
                for (j = 0; j < size; j++) {
                        scanf("%d", &a[i][j]);
                }
        }

        printf("Enter the elements for second matrix:\n");
        for (i = 0; i < size; i++) {
                for (j = 0; j < size; j++) {
                        scanf("%d", &b[i][j]);
                }
        }

        for (i = 0; i < size; i++) {
                for (j = 0; j < size; j++) {
                        if (i + j < size - 1) {
                                continue;
                        } else {
```

```
                                c[i][j] = a[i][j] + b[i][j];
                        }
                }
                printf("\n");
        }

        for (i = 0; i < size; i++) {
                for (j = 0; j < size; j++) {
                        if (i + j < size - 1) {
                                printf("__ ");
                        } else {
                                printf("%d ", c[i][j]);
                        }
                }
                printf("\n");
        }
        return 0;
}
```

Output: