Practice Set 2: Array, Structure, Function and Pointer using C programming

CS-153 Computer Programming Lab

Autumn Semester, 2016, IIT-Indore

Date: 26-08-16

Write following programs in C language.

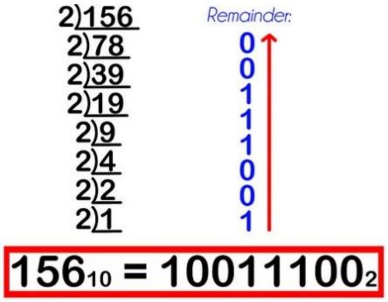
1. According to the Gregorian calendar, it was Monday on the date 01/01/1900. If any year is input through the keyboard write a program to find out what is the day on 1st January of this year.

2. Write a program using conditional to determine whether the year entered through the keyboard is a leap year or not.

3. Write a program to prints the calendar for a year taken as input.

4. Create a structure to specify data of customers in a bank. The data to be stored is: Account number, Name, Balance in account. Assume maximum of 200 customers in the bank.
(a) Write a function to print the Account number and name of each customer with balance below Rs. 100.
(b) If a customer request for withdrawal or deposit, it is given in the form:
Acct. no, amount, code (1 for deposit, 0 for withdrawal) Write a program to give a message, “The balance is insufficient for the specified withdrawal”.

5. A positive integer is entered through the keyboard. Write a function to find the binary equivalent of this number using recursion. For example, if input is 156, then binary value is 10011100.



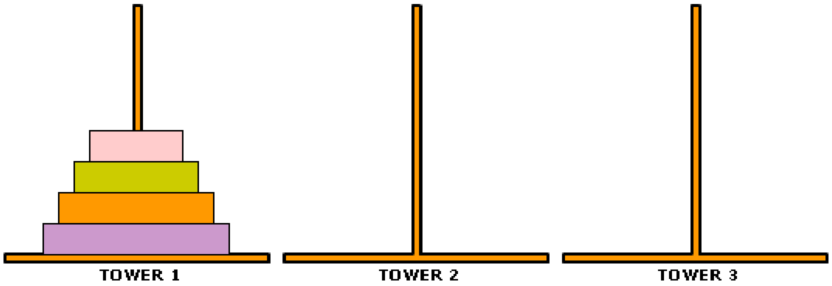
6. Write a function to compute the greatest common divisor (GCD) given by Euclid’s algorithm, exemplified for **J = 1980, K =1617** as follows:
1980 / 1617 = 1 1980 – 1 * 1617 = 363
1617 / 363 = 4 1617 – 4 * 363 = 165
363 / 165 = 2 363 – 2 * 165 = **33**
5 / 33 = 5 165 – 5 * 33 = 0
Thus, the greatest common divisor is 33. It will be a recursive solution.
Extra Point: Attempt an iterative version of this problem.

7. Write a program to compute the area of a triangle. If the lengths of the sides of a triangle are denoted by a, b, and c, then area of triangle is given by:

area = $\sqrt{S(S - a)(S - b)(S - c)}$where $S = (a + b + c) / 2$.

Write a function to compute *S*.

8. Solve the tower of Hanoi problem with 4 disks (D1, D2, D3, D4) and 3 towers (T1, T2, T3) using recursion from tower 1 (T1) to tower 3 (T3) as shown in the below figure. Display each and every move. Print how many moves do you have to make to solve a problem of N disks and 3 towers?



Remember:

Only one disk can be moved at a time.

Each move consists of taking the upper disk from one of the towers and placing it on top of another tower i.e. a disk can only be moved if it is the uppermost disk on a stack.

No disk may be placed on top of a smaller disk.

Extra Point: Attempt an iterative version of the Problem 2.

9. A positive integer is entered through the keyboard. Write a function to obtain the prime factors of this number. Also display the distinct prime factors of this number. For example, prime factors of 24 are 2, 2, 2 and 3, whereas prime factors of 35 are 5 and 7. The distinct prime factors of 24 are 2 and 3, whereas 35 are 5 and 7.

10. Write a function to compute the distance between two points and use it to develop another function that will compute the area of the triangle whose vertices are A(x1, y1), B(x2, y2),and C(x3, y3). Use these functions to develop a function which returns a value 1 if the point (x, y) is inside the triangle ABC, otherwise a value 0.

11. Write three different program to swap two numbers using

a. temporary variable
b. pointers
c. call by reference function

12. A 5-digit positive integer is entered through the keyboard, write a function to calculate sum of digits of the 5-digit number:

a. Without using recursion
b. Using recursion

```
#include <stdio.h>
int main ()                              //Monday-0
{                                        //Tuesday-1
  int c=0,m=1,i,y,p;                     //Wednesday-2
  printf("Enter a year: ");              //Thursday-3
  scanf("%d",&y);                        //Friday-4
  for(i=1991;i<=y;i++)                    //Saturday-5
  {                                      //Sunday-6
     if((i%4)==0&&(i%100)!=0||(i%400)==0) //After 7 days,c = 0;(Monday, 1 week)
      {                               //If a year leap year, next year c=c+2
         c++;                          //if(c==0) ; c = 0 ; Monday
         p=i;                          //if(c==8) ; c = 1 ; Tuesday
      }                                //c===day
      else if((i-1)==p)
         c=c+2;
     else
      c++;

      if(c==7)
         c=0;
      else if(c==8)
         c=1;
  }
  if(c==0)
   printf("MONDAY");
  else if (c==1)
   printf("TUESDAY");
  else if (c==2)
   printf("WEDNESDAY");
  else if (c==3)
   printf("THURSDAY");
  else if (c==4)
   printf("FRIDAY");
  else if (c==5)
   printf("SATURDAY");
  else if (c==6)
   printf("SUNDAY");
  return 0;
}
```

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

main() {

 int yr;
 clrscr();

 printf("Please enter the year: \n");
 scanf("%d",&yr);

 if(yr%4==0)
 printf("\nThe year is a LEAP YEAR.\n");

 else
 printf("\nThe Year is NOT A LEAP YEAR.\n");

 getch();
 return 0;

}
```

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

#define TRUE    1
#define FALSE   0

int days_in_month[]={0,31,28,31,30,31,30,31,31,30,31,30,31};
char *months[]=
{
 " ",
 "\n\n\nJanuary",
 "\n\n\nFebruary",
 "\n\n\nMarch",
 "\n\n\nApril",
 "\n\n\nMay",
 "\n\n\nJune",
 "\n\n\nJuly",
 "\n\n\nAugust",
 "\n\n\nSeptember",
 "\n\n\nOctober",
 "\n\n\nNovember",
 "\n\n\nDecember"
};


int inputyear(void)
{
 int year;

 printf("Please enter a year (example: 1999) : ");
 scanf("%d", &year);
 return year;
}

int determinedaycode(int year)
{
 int daycode;
 int d1, d2, d3;

 d1 = (year - 1.)/ 4.0;
 d2 = (year - 1.)/ 100.;
 d3 = (year - 1.)/ 400.;
 daycode = (year + d1 - d2 + d3) %7;
 return daycode;
}


int determineleapyear(int year)
{
 if(year% 4 == FALSE && year%100 != FALSE || year%400 == FALSE)
 {
  days_in_month[2] = 29;
  return TRUE;
 }
 else
 {
  days_in_month[2] = 28;
  return FALSE;
 }
}

void calendar(int year, int daycode)
{
 int month, day;
 for ( month = 1; month <= 12; month++ )
 {
  printf("%s", months[month]);
  printf("\n\nSun  Mon  Tue  Wed  Thu  Fri  Sat\n" );

  // Correct the position for the first date
  for ( day = 1; day <= 1 + daycode * 5; day++ )
  {
   printf(" ");
  }

  // Print all the dates for one month
  for ( day = 1; day <= days_in_month[month]; day++ )
  {
   printf("%2d", day );

   // Is day before Sat? Else start next line Sun.
   if ( ( day + daycode ) % 7 > 0 )
    printf("   " );
   else
    printf("\n " );
  }
   // Set position for next month
   daycode = ( daycode + days_in_month[month] ) % 7;
 }
}

int main(void)
{
 int year, daycode, leapyear;

 year = inputyear();
 daycode = determinedaycode(year);
 determineleapyear(year);
 calendar(year, daycode);
 printf("\n");
 getch();
}
```

```
#include<stdio.h>
#include<conio.h>
#define N 200

struct bank {
    int acn;
    char name[20];
    int bal;        /* defined out of main() */
    };

void main() {

 struct bank b[N];

 int i,ch,lw=100,ch2,ac,am;

 clrscr();

 for(i=0;i<N;i++) {          /* inputting customer data */

  printf("\tEnter information of customers \n");
  printf("\t******************************\n\n");

  printf("enter account no.: ");
  scanf("%d",&b[i].acn);

  printf("\n\nenter customer name: ");
  scanf("%s",&b[i].name);

  printf("\n\nenter balance: ");
  scanf("%d",&b[i].bal);

  clrscr();

 }

 clrscr();

 printf("\tEnter your choice\n");    /* further processing of transaction */
 printf("\t*****************\n\n");

 printf("1: to know whose balance is below 100.\n\n");
 printf("2: to process request or withdrawl.\n\n\n");


 scanf("%d",&ch);

 switch(ch) {

  case 1:
    clrscr();
    disp(&b);        /* displaying whose balance is below 100 */
    break;

  case 2:
   clrscr();
   printf("enter your account number: ");
   scanf("%d",&ac);

   for(i=0;i<N;i++) {

   if((b[i].acn)==ac) {
   clrscr();

   printf("\tHello %s\n",b[i].name);
   printf("\n\n");
```

```
    printf("\n\nenter your choice\n");
    printf("\n1: deposite:\n");
    printf("\n0: withdrawl:\n\n");
    scanf("%d",&ch2);

    switch(ch2) {


    case 0:

    clrscr();

   if(b[i].bal<lw) {

   printf("\n\nsorry! account balance is too low for withdrawl.\n");

   break;
   }

   else {

   printf("\n\nenter amount for withdrawl: ");
   scanf("%d",&am);

   }

   if(b[i].bal<am) {

   printf("\n\nyou don't have enough balance for withdrawl.\n");


   }


   else {

   b[i].bal=b[i].bal+am;

   printf("\n\nwithdrawl was successful.\n");

   }
   break;

    case 1:

     clrscr();

     printf("\n\nenter amount to deposite: ");
     scanf("%d",&am);

     b[i].bal=b[i].bal+am;

     printf("\n\ncash deposited successfully.\n\n");

     break;

  }
 }
 }
 }
  getch();
}

disp(struct bank *a) {
 int k;
```

```
printf("\tCustomers whose balance is below 100:\n");
printf("\t*************************************\n\n");

for(k=0;k<N;k++) {

 if((a[k].bal)<100) {

  printf("%2d\t%s\n",a[k].acn,a[k].name);

 }
}
return 0;
}
```

```
#include <stdio.h>

void swap(int*, int*);

int main()
{
  int x, y;

  printf("Enter the value of x and y\n");
  scanf("%d%d",&x,&y);

  printf("Before Swapping\nx = %d\ny = %d\n", x, y);

  swap(&x, &y);

  printf("After Swapping\nx = %d\ny = %d\n", x, y);

  return 0;
}

void swap(int *a, int *b)
{
  int temp;

  temp = *b;
  *b   = *a;
  *a   = temp;
}
```

```
 #include<stdio.h>
long toBinary(int);
int main()
{
 long binaryNo;
   int decimalNo;
 printf("Enter any decimal number: ");
   scanf("%d",&decimalNo);
 binaryNo = toBinary(decimalNo);
   printf("Binary value is: %ld",binaryNo);
 return 0;
}

long toBinary(int decimalNo)
{
 static long binaryNo, remainder, factor = 1;
 if(decimalNo != 0)
 {
  remainder = decimalNo % 2;
       binaryNo = binaryNo + remainder * factor;
       factor = factor * 10;
       toBinary(decimalNo / 2);
 }
 return binaryNo;
}
```

```
#include<stdio.h>
int main()
{
    int a,b;
    printf("Enter 2 numbers:\n");
    scanf("%d%d",&a,&b);
    printf("Greatest Common Divisor is %d",gcd(a,b)); //largest positive integer that divides the numbers without a remainder
    return 0;
}

int gcd ( int a, int b )
{
    int c;
    while ( a != 0 )   //till divisor 'a' become equal to zero
    {
        c = a;        //divisor is assigned to temporary variable c
        a = b%a;      //divisor will be changed to the mod (remainder) of 'b' dividend and divisor
        b = c;        //dividend is changed to previous divisor
    }
    return b;         //return the last dividend
}
```

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

int main()
{
    int a,b;
    printf("Enter 2 numbers:\n");
    scanf("%d%d",&a,&b);
    printf("Greatest Common Divisor is %d",gcd(a,b));

    getch();

    return 0;
}

int gcd(long a,long b)
{
    if(b==0)
        return a;
    else if(a==0)
        return b;
    else if(a==1 || b==1)
        return 1;
    else
        return gcd(b,a%b);
}
```

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

double area_of_triangle(double, double, double);

main()
{
  double a, b, c, area;

  printf("Enter the sides of triangle\n");

  scanf("%lf%lf%lf",&a,&b,&c);

  area = area_of_triangle(a, b, c);

  printf("Area of triangle = %.2lf\n", area);

  return 0;
}

double area_of_triangle( double a, double b, double c )
{
  double s, area;

  s = (a+b+c)/2;

  area = sqrt(s*(s-a)*(s-b)*(s-c));

  return area;
}
```

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

void towers(int, char, char, char);

int main()
{
   int num;

   printf("Enter the number of disks : ");
   scanf("%d", &num);
   printf("The sequence of moves involved in the Tower of Hanoi are :\n");
   towers(num, 'A', 'C', 'B');

   getch();
   getch();
   return 0;
}

/*
 * C program for Tower of Hanoi using Recursion
 */
void towers(int num, char frompeg, char topeg, char auxpeg)
{
   if (num == 1)        //terminating condition
   {   printf("\n");
       printf("\n Move disk 1 from peg %c to peg %c", frompeg, topeg); //move frompeg to topeg
       return;
   }

   towers(num - 1, frompeg, auxpeg, topeg);    //move n-1 disk frompeg to auxpeg using topeg
   printf("\n Move disk %d from peg %c to peg %c", num, frompeg, topeg);
   towers(num - 1, auxpeg, topeg, frompeg);   //move n-1 disk from auxpeg to topeg using frompeg

   printf("\n");
   getch();
}
```

```
#include<stdio.h>

void main()
{
   int x;
   printf("Input an integer\n");
   scanf("%d",&x);
   prime(x);
}

prime(int x)
{
   int a, count=0, flag_distinct=0, distinct_prime[100];

   printf("\nAll Prime Factor : ");

   for(a=2; a<=x; a++)
   {
      if (flag_distinct != a)    // flag_distinct to check distinct prime factor
      {
         distinct_prime[count] = a; // save all distinct prime factor in an array
         count=count+1;
      }

      if(x%a==0)
      {
         printf("%d ",a);  // print all prime factor
         x/=a;
         flag_distinct = a;
         a--;
      }
   }
   printf("\n\nDistinct Prime Factor :");
   for (a=0; a<count; a++)
   {
      printf(" %d ", distinct_prime[a]);   // print distinct prime factor
   }
   printf("\n");
}
```

```
#include<stdio.h>

void main()
{
    int x;
    printf("\nInput an integer\n");
    scanf("%d",&x);
    prime(x);
}

prime(int x)
{
    int a;
    for(a=2;a<=x;a++)
    {
        if(x%a==0)
        {
            printf("%d ",a);
            prime(x/a);
            break;
        }
    }
}
```

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

float distance(float x1,float y1,float x2 ,float y2)

{

    float dist;

    float dist_x = x1-x2;

    float dist_y = y1- y2;

    dist = sqrt( (dist_x * dist_x) + (dist_y * dist_y));  // calculate Euclidean distance

    return dist;

}

float area(float a,float b,float c)

{

    float s,ar;

    // Calculate area of triangle by Hero formula

    s = (a+b+c)/2;
    ar = sqrt(s*(s-a)*(s-b)*(s-c));

    return ar;

}

void main ()
{
    float x1, x2, y1, y2, a, b, c, tx1, tx2, tx3, ty1, ty2, ty3, ar_tr;

    printf ("program to calculate the distance between two points (x1,y1) and (x2, y2).");

    printf ("\n\nEnter coordinate for x1:");
    scanf ("%f", &x1);

    printf ("\nEnter coordinate for y1:");
    scanf ("%f", &y1);


    printf ("\nEnter coordinate for x2:");
    scanf ("%f", &x2);


    printf ("\nEnter coordinate for y2:");
    scanf ("%f", &y2);

    // Call distance function to calculate distance between two points
    printf ("The distance between (%f,%f) and (%f,%f) is %.2f\n\n", x1,y1,x2,y2, distance(x1,y1,x2,y2));

    printf ("Enter x and y coordinates for Vertices A\n");
    scanf("%f %f", &tx1, &ty1);

    printf ("Enter x and y coordinates for Vertices B\n");
    scanf("%f %f", &tx2, &ty2);

    printf ("Enter x and y coordinates for Vertices C\n");
    scanf("%f %f", &tx3, &ty3);
    // calculate length of each side of triangle by calling distance function
    a = distance(tx1,ty1,tx2,ty2);
    b = distance(tx1,ty1,tx3,ty3);
    c = distance(tx2,ty2,tx3,ty3);
    ar_tr = area(a, b, c);  // Calculate area of triangle by calling area function

    printf("Area of triangle is %.4f\n", ar_tr);

}
```

```c
#include <stdio.h>

int main()
{
  int x, y, temp;

  printf("Enter the value of x and y\n");
  scanf("%d%d", &x, &y);

  printf("Before Swapping\nx = %d\ny = %d\n",x,y);

  temp = x;
  x    = y;
  y    = temp;

  printf("After Swapping\nx = %d\ny = %d\n",x,y);

  return 0;
}
```

```
#include <stdio.h>

int main()
{
  int x, y, *a, *b, temp;

  printf("Enter the value of x and y\n");
  scanf("%d%d", &x, &y);

  printf("Before Swapping\nx = %d\ny = %d\n", x, y);

  a = &x;
  b = &y;

  temp = *b;
  *b   = *a;
  *a   = temp;

  printf("After Swapping\nx = %d\ny = %d\n", x, y);

  return 0;
}
```

```
#include <stdio.h>
int add_digits(int); //function declared
void main()
{
   int n, result;
    printf("Enter a 5-digit number\n");
  scanf("%d", &n);
  result = add_digits(n);
  printf("%d\n", result);
}

int add_digits(int n) //function defined
{
   int sum = 0, t, remainder=0;
  if (n == 0)
     return 0;
   t = n;
  while (t != 0)
    {
        remainder = t % 10;
        sum       = sum + remainder;
       t         = t / 10;
    }
  return sum;
}
```

```c
#include <stdio.h>
int add_digits(int); //function declared
void main()
{
   int n, result;
    printf("Enter a 5-digit number\n");
  scanf("%d", &n);
  result = add_digits(n);
  printf("%d\n", result);
}

int add_digits(int n) //function defined
{
   static int sum = 0; //static variable to keeps its value between function calls.
  if (n == 0)
     return 0;
   sum = n%10 + add_digits(n/10); //recursively calling add_digits()
  return sum;
}
```