# Institute of Engineering & Management
# Department of Computer Science & Engineering
# Data Structure Laboratory for 2nd year 3rd semester 2017
# Code: CS 392

**Date:** 26/7/17

## ASSIGNMENT-1

**Problem-1(a)**

**Problem Statement**: Program to find the number of elements in an array (Don't ask user to enter the size)

**Source code:**
```
#include <stdio.h>
void main()
{
    int arr[]={1,2,3,4,5,6,7,8,9};
    printf("The size of the array is ");
    printf("%d\n",sizeof(arr)/sizeof(int));
}
```

**Output:** The size of the array is 9

**Problem-1(b)**

**Problem Statement:** Print the alternative elements in an array.

**Source code:**
```
#include <stdio.h>
void main()
{
   int i,arr[]={0,1,2,3,4,5,6,7,8,9};
   printf("Every alternate elements are ");
   for(i=0;i<9;i=i+2)
   {
      printf("%d, ",arr[i]);
   }
}
```

**Output:** Every alternate elements are 0, 2, 4, 6, 8,

**Name:** Ranajit Roy, **Sec:** A, **Roll:** 47

## Problem-1(c)

**Problem Statement:** Increment every Element of the array by one & print incremented array (pass the whole array through a function and make necessary changes within the function body itself)

**Source code:**

```
#include <stdio.h>

void incr(int *, int);

void main()
{
   int i,arr[]={0,1,2,3,4,5,6,7,8};
   incr(arr, sizeof(arr)/sizeof(int));
}

void incr(int *arr, int len)
{
   int i;
   printf("The incremented values are ");
   for(i=0;i<len;i++)
   {
      printf("%d, ",++*(arr+i));
   }
}
```

**Output:** The incremented values are 1, 2, 3, 4, 5, 6, 7, 8, 9,

## Problem-1(d)

**Problem Statement:** Display all of the non repeated elements in an array.

**Source code:**

```
#include <stdio.h>

void main()
{
  int i, j, temp, arr[]={1,2,3,2,4,5,1,5,3},
         len=sizeof(arr)/sizeof(int);
  for(i=0;i<len-1;i++)
  {
     for(j=0;j<len-i-1;j++)
     {
        if(arr[j]>=arr[j+1])
        {
           temp=arr[j];
           arr[j]=arr[j+1];
           arr[j+1]=temp;
        }
     }
```

```
    }
    printf("the non-repeated elements are %d, ",arr[0]);
    for(i=1,temp=arr[0];i<len;i++)
    {
       if(arr[i-1]==arr[i])
          continue;
       else printf("%d, ",arr[i]);
    }
  }
```

**Output:** the non-repeated elements are 1, 2, 3, 4, 5,

## Problem-1(e)

**Problem Statement:** Segregate 0s on left side & 1s on right side of the array (traverse array only once)

**Source code:**

```
#include <stdio.h>

void main()
{
   int i, count0=0, arr[]={1,0,1,1,1,0,0,1,0,0};
   for(i=0;i<sizeof(arr)/sizeof(int);i++)
   {
      if(arr[i]==0)
         count0++;
   }
   printf("The modified array is ");
   for(i=0;i<sizeof(arr)/sizeof(int);i++,count0--)
   {
      if(count0>0)
         arr[i]=0;
      else arr[i]=1;
      printf("%d, ",arr[i]);
   }
   printf("\n");
}
```

**Output:** The modified array is 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,

## Problem-1(f)

**Problem Statement:** Pass the middle value of the array to a function to modify it by adding 10 with it

**Source code:**

```
#include <stdio.h>

void modify(int *ptr)
{
    *ptr+=10;
}

void main()
{
    int i, arr[]={1,2,3,4,5,6,7,8,9},
           size=sizeof(arr)/sizeof(int);
    printf("Before modifying ");
    for(i=0;i<size;i++)
         printf("%d, ",arr[i]);
    printf("\n");
    modify(arr-1+(size+1)/2);
    printf("After modifying ");
    for(i=0;i<size;i++)
         printf("%d, ",arr[i]);
    printf("\n");
}
```

**Output:** Before modifying 1, 2, 3, 4, 5, 6, 7, 8, 9,
After modifying 1, 2, 3, 4, 15, 6, 7, 8, 9,

## Problem-1(g)

**Problem Statement:** Sort the above array using bubble sort logic and print it.

**Source code:**

```
#include <stdio.h>
void main()
{
  int i, j, temp, arr[]={1,2,3,2,4,5,1,5,3},
            len=sizeof(arr)/sizeof(int);
  for(i=0;i<len-1;i++)
  {
      for(j=0;j<len-i-1;j++)
      {
          if(arr[j]>=arr[j+1])
          {
              temp=arr[j];
              arr[j]=arr[j+1];
              arr[j+1]=temp;
          }
      }
```

```
    }
    printf("The sorted array is ");
    for(i=0;i<len;i++){
       printf("%d, ",arr[i]);
    }
  }
```

**Output:** The sorted array is 1, 1, 2, 2, 3, 3, 4, 5, 5,

## Problem-2

**Problem Statement:** Programe to read, display, add & substract two distances. Distance must be defined using kms & meters. (Using a structure, two functions)

**Source code:**

```
#include <stdio.h>

typedef struct dts
{
    int kms, mts;
} distance;

void substract(distance *dist)

{
    if(dist[0].mts>=dist[1].mts)
    {
        dist[0].kms=dist[0].kms-dist[1].kms;
        dist[0].mts=(dist[0].mts-dist[1].mts)%1000;
    }
    if(dist[0].mts<dist[1].mts)
    {
        dist[0].kms=dist[0].kms-dist[1].kms-1;
        dist[0].mts=(dist[0].mts-dist[1].mts+1000)%1000;
    }
    printf("The result is %dkm %dmts\n", dist[0].kms,
             dist[0].mts);
}

void add(distance *dist)
{
    dist[0].kms=dist[0].kms+dist[1].kms +
                 (dist[0].mts+dist[1].mts)/1000;
    dist[0].mts=(dist[0].mts+dist[1].mts)%1000;
    printf("The result is %dkm %dmts\n", dist[0].kms,
             dist[0].mts );
}

void main()
{
    char c;
```

```
    distance dist[2];
    printf("Enter the 1st distance in kms and metres \n");
    scanf("%d %d",&dist[0].kms,&dist[0].mts);
    printf("Enter the 2st distance in kms and metres \n");
    scanf("%d %d",&dist[1].kms,&dist[1].mts);
    printf("Enter '+' to add and '-' to substract\n");
    fflush(stdin);
    scanf("%c",&c);
    switch(c)
    {
        case '+': add(dist); break;
        case '-': substract(dist); break;
        default : printf("wrong symbol\n");
    }
}
```

**Input/Output:** Enter the 1st distance in kms and metres
55
100
Enter the 1st distance in kms and metres
45
500
Enter '+' to add and '-' to substract
-
The result is 9kms 600mts

## Problem-3

**Problem statement:** A program to enter a character & determine whether it is vowel or not using switch case statement.

**Source code:**

```
#include <stdio.h>

void main()
{
    char c;
    printf("Enter a character\n");
    c=getchar();
    switch(c)
    {
        case 'a': printf("This is a vowel\n"); break;
        case 'e': printf("This is a vowel\n"); break;
        case 'i': printf("This is a vowel\n"); break;
        case 'o': printf("This is a vowel\n"); break;
        case 'u': printf("This is a vowel\n"); break;
        default :  printf("This is not a vowel\n");
    }
}
```

**Input/Output:** Enter a character
a
This is a vowel

## Problem-4

**Problem Statement:** Implement Binary Search.

**Source code:**

```
#include <stdio.h>

int n, flag=0;
void search(int *a, int max)
{
    if(n==a[max/2])
    {   flag++; return; }
    else if(max==0)
        return;
    else if(n>a[max/2])
        search( (a+(max/2)+1), (max-1)/2 );
    else if(n<a[max/2])
        search( a, (max-1)/2 );
}
void main()
{
    int a[]={1,2,4,7,9,11,15,18,19,20};
    printf("Enter the number\n");
    scanf("%d", &n);
    search( a, (sizeof(a)/sizeof(int))-1 );
    if(flag==1)
        printf("search result: found\n");
    else printf("search result: not found\n");
}
```

**Input/Output:** Enter the number
19
search result: found

## Problem-5

**Problem Statement:** A program to read a text, delete, all the semicolons it has & finally replace all ‘,’ with ‘.’

**Source code:**

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

void main()
{
    int i, count;
    char str[100];
    printf("Enter the text(<100 characters)\n");
    gets(str);
    for(i=0,count=0;i<strlen(str);i++)
    {
        if(str[i]==';')
            continue;
        else if(str[i]==',')
        {
            str[count]='.';
            count++;
        }
        else
        {
            str[count]=str[i];
            count++;
        }
    }
    str[count]='\0';
    puts(str);
}
```

**Input/Output:** Enter the text (<100 characters)
The semi-colons ‘;’ will vanish in the next line and commas ‘,’ will replace with ‘.’
The semicolons ‘’ will vanish in the next line and commas ‘.’ will repace with ‘.’

## Problem-6

**Problem Statement:** A program to copy the last n characters of a character array in another character array. Also, to convert the lower case letters into upper case while copying.

**Source code:**

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

void main()
{
    int i, n, len;
    char str[100],strcopy[100];
    printf("Enter the text(<100 characters)\n");
    gets(str);
    len=strlen(str);
    printf("Enter the number of last characters for
             copying\n");
    scanf("%d",&n);
    for(i=n;i>0;i--)
    {
        if(str[len-i]>='a'&&str[len-i]<='z')
            strcopy[n-i]=str[len-i]-('a'-'A');
        else strcopy[n-i]=str[len-i];
    }
    strcopy[n]='\0';
    puts(strcopy);
}
```

**Input/Output:**

```
Enter the text (<100 characters)
Hello world was my first program
Enter the number of last characters for copying
10
ST PROGRAM
```