Assignment 3

(1) The sum of an array is the sum of its individual elements. For example, if an array is numbers = {1, 2, 3, 4}, the sum of the array is 1+2+3+4 = 10. Function Description: Complete the function summation. The function must return the integer sum of the numbers array. **int summation(int numbers_size, int* numbers)**

```
#include <stdio.h>

int summation(int size,int *numbers){
    int i,sum=0;
    for(i=0;i<size;i++)
    {
        sum=sum+numbers[i];
    }
return sum;
}

int main(){
    int i,size,value;
    printf("please enter the array size \n");
    fflush(stdout);
    scanf("%d",&size);
    int numbers[size];
    printf("please enter values array  \n");
    fflush(stdout);
    for(i=0;i<size;i++)
        {
            scanf("%d",&numbers[i]);
        }
    value= summation(size,numbers);
    printf("The summation of the values equal: %d
\n",value);
        fflush(stdout);
    return 0;
}
```

(2) Write a function which, given a string, return TRUE if all characters are distinct and FALSE if any character is repeated.

```
#include<stdio.h>
#include<stdbool.h>
void check(char*n,int size){
    int i;
    int b=0;
    char o[]={'m','a','m','i','\0'};
    if(n[0]!=o[0])
    {
        printf("False \n");
        fflush(stdout);
    }
    else
 {
    for(i=1;i<size;i++)
  {
    if(n[i]!=o[i]){
        b=0;
    }
    else if(n[i]==n[0])
    {
        b=1;
    }
  }
    if(b==0)
    {
        printf("True");
        fflush(stdout);
    }
 }
}

int main() {
    char u[5];
```

```
    int i;
    printf("please enter a 3 letter word \n");
    fflush(stdout);
    scanf("%[^\n]%*c", u);
    check(u,4);
}
```

(5) Write a C function to return the index of **FIRST** occurrence of a number in a given array. Array index start from 0. If the item is not in the list return -1. (Linear Search Algorithm)

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

int checkplace(int*nums,int size,int num){
    int i;
    for(i=0;i<sizeof(nums);i++){
        if(nums[i]==num){
            return i;
            break;
        }
    }
    return -1;
}

int main(){
    int nums[]={2,3,4,5,44};
    int value,res;
    printf("please enter number to search for \n");
    fflush(stdout);
    scanf("%d",&value);
    res=checkplace(nums,sizeof(nums),value);
    if(res==-1)
    {
        printf("Not Found");
        fflush(stdout);
    }
```

```
    else{
        printf("Found at: %d",res);
        fflush(stdout);
    }
}
```

(6) Write a C function to return the index of **LAST** occurrence of a number in a given array. Array index start from 0. If the item is not in the list return -1. (Linear Search Algorithm) Example:

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

int checkplace(int*nums,int size,int num){
    int i,found;
    found=-1;
    for(i=0;i<10;i++)
    {
        if(num==nums[i])
        {
            found=i;
        }
    }
    if(found==-1){
    return -1;
    }
    else
    return found;
}
int main(){

    int nums[11]={2,3,4,5,6,7,6,44,5,66};
    int value,res;
    printf("please enter number to search for \n");
    fflush(stdout);
    scanf("%d",&value);
    res=checkplace(nums,sizeof(nums),value);
    if(res>=0)
    {
```

```
        printf("Found at: %d",res);
        fflush(stdout);
    }
    else{
        printf("Not Found");
        fflush(stdout);
    }

}
```

(7) Write a program that computes the nth term of the arithmetic series: 1, 3, 5, 7, 9, … Run the program to compute the 100th term of the given series.

```
#include<stdbool.h>
int compute(int val){
    int sum=0;
    sum=val+(val-1);
}
int main(){
    int val, value;
    printf("please enter number: \n");
    fflush(stdout);
    scanf("%d",&val);
    value= compute(val);
    printf("Value equal: %d \n",value);
    fflush(stdout);

}
```

(8) Write a program that computes the nth term of the geometric series: 1, 3, 9, 27, …

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

int compute(int val){
```

```
    int b,i, pow=1;
    b=val-1;
    for(i=0;i<b;i++){
    pow=pow*3;
    }
 return pow;
}
int main(){
    int val, value;
    printf("please enter number: \n");
    fflush(stdout);
    scanf("%d",&val);
    value= compute(val);
    printf("Value equal: %d \n",value);
    fflush(stdout);

}
```

(9) The sequence of numbers 1, 1, 2, 3, 5, 8, 13, … is called Fibonacci numbers; each is the sum of the preceding 2. Write a program which given n, returns the nth Fibonacci number. - with for/while

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

int fib(int val){
    int a,b,c,i;
    a=1;b=1;
    c=0;
    if((val==1)||(val==2)){
        return 1;
    }
    else
    {
        for(i=0;i<val-2;i++){
            c=a+b;
            a=b;
```

```
            b=c;
        }

    }
     return c;
}
int main(){
    int val, value;
    printf("please enter number: \n");
    fflush(stdout);
    scanf("%d",&val);
    value= fib(val);
    printf("Fibonacii Value equal: %d \n",value);
    fflush(stdout);

}
```

(9) The sequence of numbers 1, 1, 2, 3, 5, 8, 13, … is called Fibonacci numbers; each is the sum of the preceding 2. Write a program which given n, returns the nth Fibonacci number.

- with recursion

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

int fib(int val){
    if((val==1)||(val==2)){
    return 1;
}
    else{
        return fib(val-2)+fib(val-1);
    }
}
int main(){
    int val, value;
    printf("please enter number: \n");
```

```
    fflush(stdout);
    scanf("%d",&val);
    value= fib(val);
    printf("Fibonacii Value equal: %d \n",value);
    fflush(stdout);

}
```

(10) Write a function which, given a string, converts all uppercase letters to lowercase, leaving the others unchanged.

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

void down(char *arr,int size){
    int i;
    char f;
    for(i=0;i<size;i++){
        if(isupper(arr[i])){
         f=tolower(arr[i]);
         arr[i]=f;
    }
}
}
int main(){
    char arr[10];
    printf("please enter word \n");
    fflush(stdout);
    scanf("%s",arr);
    down(arr,sizeof(arr));
    printf("%s",arr);
    fflush(stdout);
}
```

(11) Write a function that prints the frequency of a certain character in a string.

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

int Count_occ(char *arr,int size,char val){
    int i,count=0;
    char f;
    for(i=0;i<size;i++){
        if(val==arr[i]){
         count++;
    }
}
    return count;
}
int main(){
    char arr[]="Ranaa";
    char o; int y;
    printf("please enter Letter \n");
    fflush(stdout);
    scanf("%c",&o);
    y=Count_occ(arr,sizeof(arr),o);
    printf("Number of Occurences of this Letter: %d
\n",y);
    fflush(stdout);
}
```

(12) Write a function to find the length of a string.

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

int length(char *arr){
    int i;
```

```
    while(arr[i]!='\0')
     {
        i++;
     }
    return i;
}
int main(){
    char arr[11];
    int y;
    printf("please enter word \n");
    fflush(stdout);
    scanf("%s",arr);
    y=length(arr);
    printf("Length of %s is : %d \n",arr,y);
    fflush(stdout);
}
```

(13) Write a function to remove all characters in a string expect alphabet.

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

void check(char *arr,int size){
    int i,j;
    for(i=0;i<size;i++){
        if(!isalpha(arr[i])){
            for(j=i;j<size-1;j++)
               arr[j]=arr[j+1];
        }
    }
}
int main(){
    char arr[11];
    printf("please enter word \n");
```

```
    fflush(stdout);
    scanf("%s",arr);
    check(arr,sizeof(arr));
    printf("After removing numbers : %s \n",arr);
    fflush(stdout);
}
```

(14) Write a function to reverse a string by passing it to a function.

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

void reve(char *arr,int size){
    int i,j=size-1;
    char arr2[6]="";
    for(i=0;i<size;i++)
    {
        arr2[i]=arr[j];
        j--;
    }
    for(i=0;i<size;i++)
    {
        arr[i]=arr2[i];
    }

}
int main(){
    char arr1[6]="";
    printf("please enter word \n");
    fflush(stdout);
    scanf("%s",arr1);
    reve(arr1,6);
    printf("Reversed word : %s \n",arr1);
    fflush(stdout);
}
```

(16) Write a C function that takes an array as input and reverse it.

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

void reve(int *arr,int size){
    int i,j=size-1;
    int arr2[6];
    for(i=0;i<size;i++)
    {
        arr2[i]=arr[j];
        j--;
    }
    for(i=0;i<size;i++)
    {
        arr[i]=arr2[i];
    }

}
int main(){
    int i, arr1[6];
    printf("please enter 6 numbers \n");
    fflush(stdout);
    for(i=0;i<6;i++)
    {
    scanf("%d",&arr1[i]);
    }
    reve(arr1,6);
    printf("Reversed numbers: ");
    for(i=0;i<6;i++)
    {
    printf(" %d ",arr1[i]);
    fflush(stdout);
    }
}
```

(18) Write a C function that return the count of the longest consecutive occurrence of a given number in an array. Example:
Array={1,2,2,3,3,3,3,4,4,4,4,3,3,3} and searching for 3 ➔ result = 4

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

int max_count(int *arr1,int size){
    int i,j,count=0,maxcount=0,search;
    for(i=0;i<size;i++)
    {
        count=0;
        search=arr1[i];
        count++;
         if(arr1[i+1]==search)
         {  count++;
            for(j=i+1;j<size-1;j++)
            {
            if(arr1[j+1]!=search)
            {
                break;
            }
            else
                count++;
            }
         }

        if(count>=maxcount)
        {
        maxcount=count;
        }
   }
return maxcount;
}

int main(){
```

```
    int arr1[]={1,1,1,1,4,3,3,3,6,6,4,4,4,4,4};
    int o;
    o= max_count(arr1,sizeof(arr1));
    printf("The maximum count of a number is
%d",o);
    fflush(stdout);

}
```

(19) Write a C function that compare between 2 arrays with the same length. It shall return 0 if the two arrays are identical and 1 if not.

```
#include<stdio.h>

int comp(int*arr1,int*arr2,int size){
    int i,found ;
    for(i=1;i<size;i++)
    {
            if(arr1[i]!=arr2[i])
            {
                return 1;
            }
    }
return 0;
}


int main(){
    int arr1[]={1,2,3,4,5,6};
    int arr2[]={1,2,3,4,5,6};
    int y;
    y=comp(arr1,arr2,sizeof(arr1));
    if(y==0){
        printf("The 2 arrays are identical");
    }
    else
```

```
            printf("The 2 arrays are different");
    }
```