1. Write a C program to demonstrate that the array elements are stored in contiguous locations.

Program:

```
#include<stdio.h>
void main()
{
    int arr[5] = {1,1,2,3,5};
    for(int i = 0; i < 5; i++)
    {
        printf("\nAddress of a%d = %u", i,
(arr+i) );
    }

}
```

Output:

```
Address of a0 = 3908053440
Address of a1 = 3908053444
Address of a2 = 3908053448
Address of a3 = 3908053452
Address of a4 = 3908053456
```

The memory address of each array item is contiguous.

2. What is the output of the following code?

```
int x[10]={1,2,3,4,5,6};
int *y;
 y=&x[0];
printf("%d", *(y+4));
```

Output:

```
    5
```

3. Consider the character array :

```
char c[10]={'C','O','R','O','N','A','!'};
 printf("%d",?);
```

Which method is used to replace the '?mark' to find the total number of characters present in it?Mention the output of the same.

Method:

```
strlen(c)
```

Output:

```
7
```

4. What is the use of free() in dynamic memory allocation?

   Dynamically allocated memory created with either *calloc()* or *malloc()* doesn't get freed on their own. You must explicitly use *free()* to release the space.

   Syntax:

```
free(ptr);
```

5. Consider the following code:

```
main()
{
   int a, *p;  // declaring the variable and
pointer
    p=&a;
    printf("%d", p+2);
    return 0;
}
```

   If the address of the variable 'a' is 790558580. Then, find p+1.

   Value of p+1 :

```
790558584
```

   Value of p+2 :

```
790558588
```

6. Execute the code given below:

```
int main()
{
     int a=90;  // declaring the variable and
pointer
     printf("%d\n", *(&a));
     return 0;}
```

Output:

```
     90
```

7. Given two matrices write a program to multiply two matrices using pointers and print the result.

Program:

```
#include <stdio.h>
void main(){
int arr1[2][2]={{5,6},{7,8} };
int arr2[2][2]={{1,2},{3,4} };
int res[2][2];
for(int i=0;i<2;i++){
    for(int j=0;j<2;j++){
        res[i][j]=0;
        for(int k=0;k<2;k++){

(*(*(res+i)+j))+=(*(*(arr1+i)+k))*(*(*(arr2+k)+j)
);
        }
    }
}
for(int i=0;i<2;i++){
    for(int j=0;j<2;j++){
        printf(" %d",res[i][j]);
    }
    printf("\n");
}
}
```

Output :

```
23 34
31 46
```

8. C – Program to illustrate the use of strcpy() and strncpy().

Program :

```
#include <stdio.h>
void main(){
char s1[25]={'H','e','l','l','o',' 
',' C','o','r','o','n','a'};
char s2[25],s3[25];
strcpy(s2,s1);
printf("strcpy : %s\n",s2);
strncpy(s3,s1,5);
printf("strncpy : %s",s3);
}
```

Output :

```
strcpy : Hello Corona
 strncpy : Hello
```

9. . Consider Daniel reads five integer values from the user as 12,21,35,64,87 using array. He wants to know the position of 87. Which algorithm may helps him to finds its position and how? Write the code for it.

Program:

```
//Binary Search
#include <stdio.h>
int binarySearch(int arr[],int size,int key){
int high=size-1,low=0,mid=0;
while(low<=high){
    mid=(low+high)/2;
    if(arr[mid]==key){
        return mid;
    }else if(key>arr[mid]){
        low=mid+1;
    }else{
        high=mid-1;
    }
```

```
}
}
void main(){
int arr[5]={12,21,35,64,87};
int size=sizeof(arr);
int key=87;//To be found
int pos=binarySearch(arr,size,key);//position of
key
printf("%d is found at position %d",key,pos);
}
```

Output:

```
87 is found at position 4
```

10. Write a program in C to find the largest element using Dynamic Memory Allocation. Consider the following sample Test Set for the execution.

Test Data :
Input total number of elements(1 to 100): 5

Number 1: 5
Number 2: 7
Number 3: 2
Number 4: 9
Number 5: 8

Expected Output :

The Largest element is : 9.00

Program:

```
#include <stdio.h>
void main(){
int *arr,n;
printf("Enter number of elements : ");
scanf("%d",&n);
arr = (int *)calloc(n, sizeof(int));
for(int i=0;i<n;i++){
    printf("\nEnter item %d :",i+1);
    scanf("%d",arr+i);
}
 for (int i = 1; i < n; ++i) {
        if (*arr < *(arr + i)){
```

```
                *arr = *(arr + i);
        }else
        {
        continue;
        }
    }
    printf("\nLargest number Entered = %d",*arr);
    }
```

Output:

```
Enter item 1 :5
Enter item 2 :7
Enter item 3 :2
Enter item 4 :9
Enter item 5 :8
Largest number Entered = 9
```

11. Consider you have declared and assigned an array variable called Arr1={1,2,2,3,3}; How can you find the unique element present in the given Arr1. Write the program for it.

Program:

```
#include <stdio.h>
void main(){
int arr[5]={1,2,2,3,3};
int index=0,key,count;
for(int i=0;i<5;i++){
    key=arr[i];
    count=0;
    for(int j=0;j<5;j++){
            if(key==arr[j]){
                count++;
            }
    }
if(count==1){
printf("Unique element : %d\n ",arr[i]);
}
}
```

```
}
```

Output:

```
Unique element : 1
```

12. How can you find the size of the following array without using sizeof() ?
    int arr[] = {10, 20, 30, 40, 50, 60};

Program:

```
#include <stdio.h>
void main(){
int arr[]={10,20,30,40,50,60};
int size =(&arr)[1]-arr;
printf("size of array = %d",size);
}
```

Output:

```
size of array = 6
```

13. Write the program to count the number of digits present in a given number using function recursive method.

Program:

```
#include <stdio.h>
int countDigit(int num,int count){
if((num/10)!=0){
    num/=10;
    count++;
    return countDigit(num,count);
}else{
count++;
return count;
}
}
void main(){
int res = countDigit(12345678,0);
printf("Number of digits = %d ",res);
```

```
}
```

Output:

```
Number of digits = 8
```

14. Consider the below code,

```
int main()
{
int x[]={?,?,?,?,?};
int *ptr;
ptr=&x[?];
printf("*(ptr) = %d \n", *(ptr));
printf("*(ptr+1) = %d \n", *(ptr+1));
printf("*(ptr+2) = %d \n", *(ptr+2));
}
```

Program:

```
#include <stdio.h>
int main(){
int x[]={3,2,6,9,1};
int *ptr;
ptr=&x[0];
printf("*(ptr) = %d\n",*(ptr));
printf("*(ptr+1) = %d\n",*(ptr+1));
printf("*(ptr+2) = %d\n",*(ptr+2));
}
```

Output:

```
*(ptr) = 3
*(ptr+1) = 2
*(ptr+2) = 6
```

15. Write a program in C to count a total number of duplicate elements in an given array.

Program:

```
#include <stdio.h>
void main(){
int arr[30],n;
printf("Enter number of elements : ");
scanf("%d",&n);
for(int i=0;i<n;i++){
    printf("\nEnter item %d : ",i+1);
    scanf("%d",&arr[i]);
}
int index=0,key,count,prev=0;
for(int i=0;i<n;i++){
    key=arr[i];
    count=0;
    for(int j=0;j<n;j++){
            if(key==arr[j]){
                count++;
            }
    }
if(count>1 && arr[i]!=prev){
printf("Duplicate element : %d\n ",arr[i]);
prev=arr[i];
}
}
}
```

Output:

```
Enter number of elements : 5
Enter item 1 : 1
Enter item 2 : 2
Enter item 3 : 3
Enter item 4 : 1
Enter item 5 : 5
Duplicate element : 1
```

16. Given two matrices, write a program to compute the subtraction of those matrices.

Program:

```
#include <stdio.h>
void main(){
```

```
int arr1[2][2]={{5,6},{7,8} };
int arr2[2][2]={{1,2},{3,4} };
int res[2][2];
for(int i=0;i<2;i++){
    for(int j=0;j<2;j++){
        res[i][j]=arr1[i][j]-arr2[i][j];
    }
}
for(int i=0;i<2;i++){
    for(int j=0;j<2;j++){
        printf(" %d",res[i][j]);
    }
    printf("\n");
}
}
```

Output:

```
4 4
4 4
```

17. Write the program for the given concept using function. If a given input is "@123!," then your function should return "!321@".

Program:

```
#include <stdio.h>
void stringRev(char arr[]){
for(int i=(sizeof(arr)/sizeof(arr[1]))-1;i>-1;i--
)
    printf("%c",arr[i]);
}
void main(){
char arr[]="@123!hjk";
stringRev(arr);
}
```

Output:

```
kjh!321@
```

18. Write the c program for finding the power of any given number using function recursive. Use the function GetPower() with two arguments ‘base’ and ‘power’.

Program:

```
#include <stdio.h>
int getPower(int base , int power){
int count=0,sum=1;
while(count<power){
    sum*=base;
    count++;
}
return sum;
}
void main()
{
int result =getPower(2,5);
printf("2^5 = %d",result);
}
```

Output:

```
2^5 = 32
```

19. Write the program to find greatest of three numbers.

Program:

```
#include <stdio.h>
int Big(int,int,int);
void main()
{
int great =Big(10,40,20);
printf("Greatest = %d",great);
}
int Big(int a,int b,int c)
{
if(a>b&&a>c){
    return a;
}else if(b>a&&b>c){
    return b;
}else{
return c;
}
}
```

Output:

```
Greatest = 4
```

20. Compute the output of the program.

Program:

```
#include <stdio.h>
int main() {
        char *ptr;
        char string[] = "happy to see you all";
        ptr = string;
        ptr += 6;
        printf("%s",ptr);
        return 0;
}
```

Output:

```
to see you all
```

Reason:

The variable ptr is assigned with the memory address of string, upon adding ptr+6 the value of ptr variable changes to the sixth index value of string ,so while printing ptr the output will be the string starting from index 6 and till the end of the string.