Array

1

```
#include<iostream>
using namespace std;
int main(){
    int a[100]={0};
    for(int i=0;i<100;i++ ){
    cout<<a[i];
    }
}
```

2

```
#include <iostream>

using namespace std;
void printArray(int *arr,int n){
    cout<<"in the function "<<sizeof(arr)<<endl;
    cout<<"in the function "<<sizeof(int)<<endl;
    cout<<"in the function "<<n<<endl;
}
int main(){
    int arr[]={1,2,3,4,5,6,7,8,9,10};
    int n = sizeof(arr)/sizeof(int);
    cout<<"in the main function "<<sizeof(arr) <<endl;
    cout<<"in the main function "<<sizeof(int)<<endl;
    printArray(arr,n);
    return 0;
}
```

3

```
//linear search
#include <iostream>
using namespace std;
void linear_search(int *arr,int n,int key){
    for(int i=0;i<n;i++){
        if(arr[i]==key){
            cout<<"key is found in the array"<<endl;
            break;
        }
    }
    cout<<"key is not found in the array";
}
int main(){
    int n,key;
    cin>>n;
    int arr[n];
```

```
    for(int i=0;i<n;i++){
        cin>>arr[i];
    }
    cin>>key;
    linear_search(arr,n,key);
}
```

4

```
//binary search
#include <iostream>
using namespace std;
int linear_search(int *arr,int n,int key){
    int start=0;
    int end=n-1;
    while(start<=end){
        int mid=(start+end)/2;
        if(arr[mid]==key){
            return 1;
        }else if(arr[mid]>key){
            end=mid-1;
        }else {
            start=mid+1;
        }
    }
    return -1;
}
int main(){
    int n,key;
    cin>>n;
    int arr[n];
    for(int i=0;i<n;i++){
        cin>>arr[i];
    }
    cin>>key;
    int ans=linear_search(arr,n,key);
    if(ans==1){
        cout<<"key is found"<<endl
    };
    else{
        cout<<"key is not found"<<endl;
    }
}
```

5

```
//reverse the array
#include <iostream>
```

```cpp
using namespace std;
void reverse_array(int arr[],int n){
    int s=0;
    int e=n-1;
    while(s < e){
        swap(arr[s],arr[e]);
        s++;
        e--;
    }
}
int main(){
    int arr[]={4,5,6,7,8,9,10,11,12};
    int n=sizeof(arr)/sizeof(int);
    reverse_array(arr,n);
    for (int i = 0; i < n; ++i)
    {
        cout<<arr[i]<<endl;
    }
}
```

6

```cpp
//printing the pairs
#include <iostream>
using namespace std;
void printing_pairs(int arr[],int n){
    for (int i = 0; i < n; ++i)
    {
        for (int j = i+1; j < n; ++j)
        {
            cout<<"("<<arr[i]<<" , "<<arr[j]<<")"<<endl;
        }
    }
}
int main(){
    int arr[]={4,5,6,7,8,9,10,11,12};
    int n=sizeof(arr)/sizeof(int);
    printing_pairs(arr,n);
}
```

7

```cpp
//printing the sub array and calulate the sum of that
//and find out the largest sum
#include <iostream>
using namespace std;
void printingSubArray(int *arr, int n){
    for (int i = 0; i < n; ++i)
```

```
    {
        for (int j = i; j < n; ++j)
        {
            for (int k = i; k <= j; ++k)
            {
                cout<<arr[k]<<",";
            }
            cout<<endl;
        }
    }
}
void sunOfPairs_largestSum1(int *arr, int n){
    int large=0;
    for (int i = 0; i < n; ++i)
    {
        for (int j = i; j < n; ++j)
        {
            int sum=0;
            for (int k = i; k <= j; ++k)
            {
                sum=sum + arr[k];
                if(k==j){
                    cout<<sum;
                }
            }
            cout<<" ";
            if(sum>large){
                large=sum;
            }
        }
        cout<<endl;
    }
    cout<<large;
}
//prefix sum approch 0(n^2)
void sunOfPairs_largestSum2(int *arr, int n){
    //prefix sums
    int prefix[n]={0};
    prefix[0]=arr[0];
    for (int i = 1; i < n; ++i)
    {
        prefix[i]=prefix[i-1]+arr[i];
    }
    //largest sum logic
    int large=0;
    for (int i = 0; i < n; ++i)
    {
        for (int j = i; j < n; ++j)
```

```
        {
            int sum=i>0 ? prefix[j]-prefix[i-1] : prefix[j];
            if(sum>large){
                large=sum;
            }
        }
    }
    cout<<large;
}

int main(){
    int arr[]={4,5,6,7,8,9,10,11,12};
    int n=sizeof(arr)/sizeof(int);
    printingSubArray(arr,n);
    sunOfPairs_largestSum2(arr,n);
}
```

8

```
#include<iostream>
using namespace std;
int maximum_subarray_sum(int *arr,int n){
    int cs = 0;
    //current sum
    int largest = 0;
     //largest sum
    for (int i = 0; i < n; ++i)
    {
        cs=cs+arr[i];
        if(cs < 0){
            cs = 0;
        }
        largest = max(largest, cs);
    }
    return largest;
}
int main()
{
    //array containing
    int arr[]={4,5,6,7,8,9,10,11,12};
    int n = sizeof(arr)/sizeof(int);
    cout<< maximum_subarray_sum(arr,n) <<endl;
}
```