①

```
#include<iostream>
#include<time.h>
#include<stdlib.h>
using namespace std;
void SWAP (int*p, int LP, int SP, int s)
{   int temp1, temp2;
    for(int i=0; i<s; i++)
     {  if(p[i]==LP)
         { temp1=i;
         }
        if(p[i]==SP)
        { temp2=i;
        }
      }
    p[temp1]=SP;
    p(temp2]=LP;
}
void cum(int*a, int s)
{   int help[s];
    help[0]=a[0]
    for(int i=1; i<s; i++)
     {  ~~help~~ help[i]=a[i]+help[i-1];
     }
```

```
    for(int k=l ; k<s ; k++)
    {  a[k] = help[k];
    }
}
Void print(int *p , int s)
{   cout<<endl;
    for(int i=0 ; i<s; i++)
    { cout<<p[i]<<" ";
    }
}
int main()
{   int n, LP, SP;
    cout<<"Enter the size of array: ";
    cin>>n ;
    int * A = new int [n];
    Srand(time(0));
   for(int i=0 ; i<n ; i++)
   {  A[i] = rand()%50;
      cout<< A[i]<<" ";
   }
  LP = A[0];
 SP = A[0];
```

```
for(int k=0 ; k<n ; k++)
{  if(LP<A[k])
   {
      LP=A[k];
   }
   if(SP>A[k])
   {  SP=A[k];
   }
}
cout<<"\nLP "<<LP;
cout<<"\nSP "<<SP;
SWAP(A, LP, SP, n);
cout << "\nAfter swap: ";
Print(A, n);
cout << endl;
cum(A,n);
cout << "\n cumulative sum : ";
Print(A,n);
 delete [] A;
 return 0;
}
```

②

```
#include<iostream>
Using namespace std;
int sum = 1;
int fib(int a, int b, int n)
{
    if(a==0)
    { cout<< 0 << " " << 1 << " ";
    }
  int temp;
  temp=a+b;
  a=b;
  b=temp;
  cout<< temp << " ";
  Sum = sum + temp;
   n--;
  if(n-2>0)
  { fib(a,b,n);
   }
   return sum;
}
int main()
{  int s;
   s=fib(0,1,15);
  cout<<"\n The summation is "<<s;
  return 0;
}
```

③
```
#include<iostream>
#include<stack>
Using namespace std;
void print_stack(stack<int> a)
{   stack<int> b = a;
    while(!b.empty())
    {   cout<< b.top()<<"\n";
        b.pop();
    }
}
int main()
{   int inp,x;
    stack<int> b;
    stack<int> s;
  for(int i=0; i<10 ;i++)
  {   cout<< "Enter a stack value: ";
      cin>> inp;
      s.push(inp);
  }
  cout<<"The initial stack: \n";
  print_stack(s);
  cout<<"Enter a value upto which
  cout<< "Enter the element to pop:";
  cin>> x;
  cout<< "The stack after delete: \n";
```

```
while (!s.empty())
{   if(!(s.top()>=x))
    {  b.push(s.top());
    }
    s.pop();
}
while(!b.empty())
{   s.push(b.top());
    b.pop();
}
print_stack(s);
 return 0;
}
```