Pointers

Ex.No.: **Date:**

ReverseaList

ProblemStatement:

Givenanarrayofintegers,reversethegivenarrayinplaceusinganindexandlooprather than a built-in function.

Example
arr=[1,3, 2,4, 5]
Returnthearray[5,4,2,3,1]whichisthereverseoftheinputarray.

FunctionDescription
CompletethefunctionreverseArrayintheeditorbelow.

reverseArray has the following parameter(s):
int arr[n]: an array of integers

Return
int[n]:thearrayinreverseorder

Constraints
1≤n≤100
0<arr[i]≤100

InputFormatForCustom Testing
Thefirstlinecontainsaninteger,n,thenumberofelementsinarr.
Eachlineiofthensubsequentlines(where0≤i<n)containsaninteger,arr[i].

Sample Input For Custom Testing
5
1
3
2
4
5

Sample Output
5
4
2
3
1

Explanation
Theinputarrayis[1,3,2,4,5],sothereverseoftheinputarrayis[5,4,2,3,1].

Program:

```
int* reverseArray(int arr_count, int *arr, int *result_count)
    *result_count=arr_count;
    for(int i=0;i<arr_count/2;i++)
    {
        int temp=arr[i];
        arr[i]=arr[arr_count-i-1];
        arr[arr_count-i-1]=temp;
    }
    return arr;
}
```

Test	Expected	Got	
int arr[] = {1, 3, 2, 4, 5};	5	5	✓
int result_count;	4	4	
int* result = reverseArray(5, arr, &result_count);	2	2	
for (int i = 0; i < result_count; i++)	3	3	
printf("%d\n", *(result + i));	1	1	

Ex.No.: **Date:**

CutThem All

ProblemStatement:

Anautomatedcuttingmachineisusedtocutrodsintosegments.Thecuttingmachinecan only hold a rod of minLength or more, and it can only make one cut at a time. Given the array lengths [] representing the desired lengths of each segment, determine if it is possible to makethe necessary cuts using thismachine. The rod ismarked into lengths already, in the order given.

Example
n = 3
lengths=[4,3,2]
minLength=7

Therodisinitiallysum(lengths)=4+3+2=9unitslong.Firstcutoffthesegmentoflength 4 + 3 = 7 leaving a rod 9 - 7 = 2. Then check that the length 7 rod can be cut into segmentsof lengths 4 and 3. Since 7 is greater than or equal to minLength = 7, the final cut can be made. Return "Possible".

Example
n = 3
lengths=[4,2,3]
minLength=7

Therodisinitiallysum(lengths)= 4+ 2+ 3=9unitslong.Inthiscase, theinitialcutcan beoflength4or4+2=6.Regardlessofthelengthofthefirstcut,theremainingpiece will be shorter than minLength. Because n - 1 = 2 cuts cannot be made, the answer is "Impossible".

FunctionDescription
CompletethefunctioncutThemAllintheeditorbelow.

cutThemAllhasthefollowingparameter(s):
intlengths[n]:thelengthsofthesegments,inorder
intminLength:theminimumlengththemachinecanaccept

Returns
string:"Possible"ifalln-1cutscanbemade.Otherwise,returnthestring"Impossible".

Constraints
- 2≤n≤105
- 1≤t≤109
- 1≤lengths[i]≤109
- Thesumoftheelementsoflengthsequalstheuncutrodlength.

InputFormatForCustom Testing

Thefirstlinecontainsaninteger,n,thenumberofelementsinlengths.
Each line iofthe n subsequent lines(where0 ≤i < n)contains an integer, lengths[i].
Thenextlinecontainsaninteger,minLength,theminimumlengthacceptedbythe machine.

SampleInputForCustomTesting

STDIN		Function

4---	→	lengths[]sizen=4
3	→	lengths[]=[3,5,4,3]
5		
4		
3		
9	→	minLength=9

SampleOutput
Possible

Explanation

Theuncutrodis3+5+4+3=15unitslong.Cuttherodintolengthsof3+5+4 =12and3.
Thencutthe12-unitpieceintolengths3and5+4=9.
Theremainingsegmentis5+4=9unitsandthatislongenoughtomakethefinal cut.

Program:

```
char* cutThemAll(int lengths_count, long *lengths, long minLe
    long t=0,i=1;
    for(int i=0;i<=lengths_count-1;i++)
    {
        t+=lengths[i];
    }
    do
    {
        if(t-lengths[lengths_count-i-1]<minLength)
        {
            return"Impossible";
        }
        i++;
    }while(i<lengths_count-1);
    return "Possible";
}
```

Test	Expected	Got	
long lengths[] = {3, 5, 4, 3}; printf("%s", cutThemAll(4, lengths, 9))	Possible	Possible	✓
long lengths[] = {5, 6, 2}; printf("%s", cutThemAll(3, lengths, 12))	Impossible	Impossible	✓