

RHP ASSIGNMENT

1. Josephus problem

```
#include <iostream>
```

```
#include <bits/stdc++.h>
```

```
#include <vector>
```

```
using namespace std;
```

```
int josephus(vector<int> vec, int k, int ind)
```

```
{
```

```
    if (vec.size() == 1)
```

```
    {
```

```
        return vec[0];
```

```
    }
```

```
    ind = (ind + k) % vec.size();
```

```
    vec.erase(vec.begin() + ind);
```

```
    return josephus(vec, k, ind);
```

```
}
```

```
int main()
```

```
{
```

```
    int n, k;
```

```
cout << "Enter the number of people: ";
cin >> n;
cout << "Enter the number to skip people: ";
cin >> k;
k--;
vector<int> vec;
int ind = 0;
for (int i = 1; i <= n; i++)
{
    vec.push_back(i);
}

cout << "The person at the position " << josephus(vec, k, ind) << "
survives";
}
```

2. apples

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
int main()
{
```

```
cout<<"Enter number of people with apples: "<<endl;
int N, avg, sum=0, count=0;
cin>>N;
int arr[N];
for(int i=0; i<N; i++)
{
    cin>>arr[i];
    sum+=arr[i];
}
avg=sum/N;
sort(arr,arr+N, greater<int>());
for(int i=0; i<N; i++)
{
    if(arr[i]>avg)
    {
        for(int j=i+1; j<N; j++)
        {
            while(arr[j]!=avg && arr[i]!=avg)
            {
                arr[j]++;
                arr[i]--;
                count++;
            }
        }
    }
}
```

```

        }
        if(arr[i]==avg)
            break;
    }
}
}
cout<<"\nNo. of operations to have equal apples: "<<count;
}

```

3. array(convert into palindromic then)

```

#include<iostream>
#include <bits/stdc++.h>
#include<string.h>
using namespace std;

int findMinInsertions(char str[], int l, int h)
{
    if (l > h)
        return INT_MAX;
    if (l == h)
        return 0;
}

```

```
if (l == h - 1)
```

```
    return (str[l] == str[h]) ? 0 : 1;
```

```
    return (str[l] == str[h]) ? findMinInsertions(str, l + 1, h - 1) :
```

```
(min(findMinInsertions(str, l, h - 1), findMinInsertions(str, l + 1, h)) + 1);
```

```
}
```

```
int main()
```

```
{
```

```
    cout<<"Enter the string: ";
```

```
    char str[1001];
```

```
    cin>>str;
```

```
    cout<<"\nThe no. of operations required to make it palindrome: ";
```

```
    cout << findMinInsertions(str, 0, strlen(str) - 1);
```

```
    return 0;
```

```
}
```

4. substring

```
n = int(input("Enter the Weight of the string: "))
```

```
s = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

```
c = 4
```

```
d={}
d['A']=1
d['B']=3
for i in range(2,len(s)):
    d[s[i]]=d[s[i-1]]*c
    if d[s[i]]>n:
        del d[s[i]]
        break
    c+=1
l=list(d.keys())
print("The character used are :")
print(l)
print("The string equal to given weight is: ")
res=""
for i in range(len(l)-1,-1,-1):
    r=n//d[l[i]]
    res+=l[i]*r
    n=n%d[l[i]]
    if n==0:
        break
print(res[::-1])
```

5. swap elements in an array

```
#include<iostream>

using namespace std;

int main()
{
    int N;
    cin>>N;
    int arr[N];
    for(int i=0; i<N; i++)
    {
        cin>>arr[i];
    }
    cout<<"Before swapping"<<endl;
    for(int i=0; i<N; i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<"\nAfter swapping"<<endl;
    for(int i=0; i<N-1; i+=2)
    {
        swap(arr[i],arr[i+1]);
    }
```

```
for(int i=0; i<N; i++)  
{  
    cout<<arr[i]<<" ";  
}  
}
```

6. character to read

```
#include <iostream>  
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()  
{  
    cout<<"Enter the string: ";  
    string s;  
    cin >> s;  
    int c[26] = {0}, max = 0;  
    for (int i = 0; s[i]; i++)  
    {  
        c[s[i] - 'a']++;  
    }
```



```
}  
cout<<"Words that are difficult learn: "<<endl;  
for (int i = 25; i >= 0; i--)  
{  
    if (c[i] > max)  
        max = c[i];  
    if (c[i] == 0)  
    {  
        cout << char(i + 'a');  
    }  
}  
int k = 1;  
while (k <= max)  
{  
    for (int i = 25; i >= 0; i--)  
    {  
        if (c[i] == k)  
        {  
            cout << char(i + 'a');  
        }  
    }  
    k++;  
}
```

```
}  
}
```

7. Staircase problem

```
#include <iostream>  
#include <bits/stdc++.h>  
using namespace std;  
int main()  
{  
    int N, n1=0, n2=1, n3=0;  
    cout<<"Enter no. of steps:";  
    cin>>N;  
    while(N>0)  
    {  
        n3=n1+n2;  
        n1=n2;  
        n2=n3;  
        N--;  
    }  
    cout<<"\nThe no. ways: "<<n3;  
}
```

8. Pendulum problem

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
int main()
{
    cout<<"Enter the size and elements:";
    int N;
    cin>>N;
    int arr[N], pendulam[N], index=N-1, penLeft=0, penRight=N-1;
    for(int i=0; i<N; i++)
    {
        cin>>arr[i];
    }
    sort(arr, arr+N);
    while(index>=0)
    {
        if(index>=0)
            pendulam[penRight--]=arr[index--];
        if(index>=0)
```

```

        pendulam[penLeft++]=arr[index--];
    }
    cout<<"Pendulam pattern :\n";
    for(int i=0; i<N; i++)
    {
        cout<<pendulam[i]<<" ";
    }
}

```

9. Selection of cities

```

#include <iostream>
#include <cstring>
using namespace std;
char input[100], output[100];
void Comb(char *input, int index, char *output, int outLen)
{
    if (input[index] == '\0')
    {
        output[outLen-1] = '\0';
        cout << output << endl;
        return;
    }
}

```

```

    }
    output[outLen] = input[index];
    output[outLen + 1] = ',';
    Comb(input, index + 1, output, outLen + 2);
    if (input[index + 1] != '\0')
        Comb(input, index + 1, output, outLen + 1);
}

int main()
{
    cout<<"Enter the String: ";
    cin>>input;
    output[0] = '\0';
    cout<<"\nCities that can be visited are:"<<endl;
    Comb(input, 0, output, 0);
    return 0;
}

```

10.Maximum profit of sales

```

#include <iostream>
using namespace std;

```

```
int sum=0, c=0;
void stockBuyAndSell(int arr[], int n)
{
    int i = 0;
    while (i < n - 1)
    {
        while ((i < n - 1) && arr[i + 1] <= arr[i])
        {
            i++;
        }
        if (i == n - 1)
        {
            break;
        }
        int minima = i++;
        while ((i < n) && arr[i] >= arr[i - 1])
        {
            i++;
        }
        int maxima = i - 1;
        sum += arr[maxima]-arr[minima];
    }
}
```

```
    cout << "(" << minima << " " << maxima << ") Profit: "<<arr[maxima]-  
arr[minima]<<endl;
```

```
    c++;
```

```
}
```

```
if (c == 0)
```

```
{
```

```
    cout << "No Profit";
```

```
}
```

```
}
```

```
int main()
```

```
{
```

```
    int n;
```

```
    cin >> n;
```

```
    int arr[n];
```

```
    for (int i = 0; i < n; i++)
```

```
    {
```

```
        cin >> arr[i];
```

```
    }
```

```
    stockBuyAndSell(arr, n);
```

```
    if(c!=0)
```

```
        cout<<"\nTotal Profit: "<<sum;
```

```
return 0;
```

```
}
```