
Question 1:

```
1 //| 106119100 Rajneesh Pandey
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 void printSubStr(
6     string str, int low, int high)
7 {
8     for (int i = low; i <= high; ++i)
9         cout << str[i];}
10
11 int longestPalSubstr(string str)
12 {
13     int n = str.size();
14     bool table[n][n];
15     memset(table, 0, sizeof(table));
16     int maxLength = 1;
17     for (int i = 0; i < n; ++i)
18         table[i][i] = true;
19     int start = 0;
20     for (int i = 0; i < n - 1; ++i) {
21         if (str[i] == str[i + 1]) {
22             table[i][i + 1] = true;
23             start = i;
24             maxLength = 2;
25         }
26     }
27     for (int k = 3; k <= n; ++k) {
28         for (int i = 0; i < n - k + 1; ++i) {
29             int j = i + k - 1;
30             if (table[i + 1][j - 1] && str[i] == str[j]) {
31                 table[i][j] = true;
32                 if (k > maxLength) {
33                     start = i;
34                     maxLength = k;
35                 }
36             }
37         }
38     }
39     cout << "Longest palindrome substring is: ";
40     printSubStr(str, start, start + maxLength - 1);
41     return maxLength;
42 }
43
44 int main()
45 {
46     int t; cin >> t;
47     while(t--){ string str; cin >> str;
48         longestPalSubstr(str);
49         cout << endl; }
50     return 0; }
```

Output/Input

4
babad
cbbd
a
ac

Copy

Run Run+URL (Generates URL as well)

Run Program(Ctrl+Enter)

Time(sec) : 0Memory(MB) : 3.3697058575439

Output:Copy

Longest palindrome substring is: bab
Longest palindrome substring is: bb
Longest palindrome substring is: a
Longest palindrome substring is: a

Question 2:

```
1 // 106119100 Rajneesh Pandey
2 #include<bits/stdc++.h>
3 using namespace std;
4
5 int dp[1001][1001];
6
7 int knapSack(int W, int wt[], int val[], int n)
8 {
9     for(int i=0;i<=n;i++)
10    {
11        for(int j=0;j<=W;j++)
12            {if(i==0 or j==0)
13                dp[i][j] = 0;}
14    }
15    for(int i=1;i<=n;i++)
16    {
17        for(int j=1;j<=W;j++)
18        {
19            if(wt[i-1]<=j)
20                dp[i][j] = max(val[i-1]+dp[i-1][j-wt[i-1]], dp[i-1][j]);
21            else
22                dp[i][j] = dp[i-1][j];
23        }
24    }
25    return dp[n][W];}
26
27 int main()
28 {
29     int t;
30     cin>>t;
31     while(t-->0)
32     {
33         int n, w;
34         cin>>n>>w;
35         int val[n];
36         int wt[n];
37         for(int i=0;i<n;i++)
38             cin>>val[i];
39         for(int i=0;i<n;i++)
40             cin>>wt[i];
41         cout<<knapSack(w, wt, val, n)<<endl;
42     }
43     return 0;
44 }
```

Input/Output

```
1
5 15
1 20 3 14 100
2 10 3 6 18
```

Copy

Run

Run+URL (Generates URL as well)

Run Program(Ctrl+Enter)

Time(sec) : 0

Memory(MB) : 7.1004224981689

Output:

Copy

24