BUBT | BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY

Committed to Academic Excellence

Assignment

Course Title : Algorithm Lab

Course Code : 242
Assignment No : 05 & 06

Submitted By:

Name: Rabbi Hasan
ID No: 21225103162
Intake: 49
Section: 10

Submitted To:

Name: Faria Binte Kader
Department of: CSE
Bangladesh University of Business & Technology.

Date of Submission: 23-09-2023

Signature of Teacher

Lab Task 05

Ans to the question number 01

```
// Answer to the question no: 01 //
#include<bits/stdc++.h>
using namespace std;

bool cmp(const pair<int,int> &a,const pair<int,int> &b){
 return (a.second < b.second);
}

int main() {
    cout << "Given Item No: ";
    int n; cin >> n;
    int a[n+1],b[n+1];
    for(int i = 0; i < n; i++){
    cin >> a[i];
    }for(int i = 0; i < n; i++){
    cin >> b[i];
    }

    vector< pair<int, int> >v;
    for(int i = 0; i < n; i++){
     v.push_back({a[i] , b[i]});
    }
    sort(v.begin(), v.end(),cmp);

    int ans = 1,  prev = b[0];
    for (int i = 1; i < n; ++i) {
        if (a[i] >= prev) {
            ans++;
            prev = b[i];
        }
    }
    cout<< "\nMaximum Activity: " << ans <<'\n';
    return 0;
}
```

Ans to the question number 02

```
// Answer to the question no: 02 //
#include <bits/stdc++.h>
using namespace std;

bool compare(pair<int, int> item1, pair<int, int> item2) {
    double R1 = (double)item1.first / item1.second;
    double R2 = (double)item2.first / item2.second;
    return R1 > R2;
}

double fractionalKnapsack(int W, vector<pair<int, int>>& items) {

    double total = 0.0;
    for (auto& it : items) {
        if (W >= it.second) {
            total += it.first;
            W -= it.second;
        } else {
            double fraction = (double)W / it.second;
            total += fraction * it.first;
            break;
        }
    }

    return total;
}

int main() {
    int n=3;//cin >> n;
    vector<pair<int, int>> items;
    for(int i=0; i<n; i++){
     int p,w;
     cin >> p >> w;
     items.push_back({p, w});
    }
    cout << "Enter Max Weight: ";int W; cin >> W;

    sort(items.begin(), items.end(), compare);
    double ans = fractionalKnapsack(W, items);
    cout << "\nThe max Profit: " << ans << "\n";

    return 0;
}
```

Ans to the question number 03

```
// Answer to the question no: 03 //
#include<bits/stdc++.h>
using namespace std;

bool compare(pair<int, int> item1, pair<int, int> item2) {
    double R1 = item1.first / item1.second;
    double R2 = item2.first / item2.second;
    return R1 > R2;
}

int knapsack(int W, vector<pair<int, int>> &items){
    int total= 0;
    int mxcapacity = W;
    for (auto &it: items){
        if (mxcapacity >= it.second) {
            total += it.first;
            mxcapacity -= it.second;
        }
        else{
          break;
        }
    }
    return total;
}

int main() {
    int n=3; //cin >> n;
    vector<pair<int, int>> items;
    for(int i=0; i<n; i++){
     int p,w;
     cin >> p >> w;
     items.push_back({p, w});
    }
    cout << "Enter Max Weight: ";int W; cin >> W;

    sort(items.begin(), items.end(), compare);
    int ans = knapsack(W, items);
    cout << "The max Profit: " << ans << "\n";

    return 0;
}
```

Lab Task 06

Ans to the question number 01

```
// Answer to the question no: 01 //
#include<bits/stdc++.h>
using namespace std;

int knapsack(vector<int>& W , vector<int>& P , int x) {
    int n = W.size();
    vector<vector<int>> dp(n + 1, vector<int>(x + 1, 0));

    for (int i = 1; i <= W.size(); ++i) {
        for (int j = 1; j <= x; ++j) {
            if (W[i - 1] > j){
                dp[i][j] = dp[i - 1][j];
                }
            else{
               dp[i][j] = max(dp[i - 1][j] , dp[i - 1][j - W[i - 1]] + P[i - 1]);
                }
        }
    }
    return dp[n][x];
}

int main() {
    int n=3; //cin >> n;
    vector<int> w(n);
    cout << "Enter Weights: ";
    for(int i=0; i<n; i++){
     cin >> w[i];
    }
    vector<int> p(n);
    cout << "\nEnter Profits: ";
    for(int i=0; i<n; i++){
     cin >> p[i];
    }
    cout << "\nEnter Max Capacity: ";int c; cin >> c;

    int Ans = knapsack(w, p, c);
    cout << "\nMaximum profit: " << Ans << endl;

    return 0;
}
```

Ans to the question number 02

```
// Answer to the question no: 02//
#include <bits/stdc++.h>
using namespace std;

const int N=100;
int dp[N][N];

int main()
{
    int l;
    cin >> l;
    int c[5],p[5];
    for(int i=0; i<l; i++){
    cin>>p[i];
     }

    for(int i=0; i<=l; i++){
     for(int j=0; j<=l; j++){
      if(i==0 || j==0){
       dp[i][j]=0;
      }
      else{
       if(i==1){
        dp[i][j] = j*p[i-1];
       }
       else{
         if(i>j){
           dp[i][j]=dp[i-1][j];
         }
         else{
          dp[i][j]=max(p[i-1] + dp[i][j-i], dp[i-1][j]);
         }
       }
      }
     }
    }
  cout<<"MAX ANS: "<<dp[l][l]<<'\n';
 return 0;
}
```

Ans to the question number 03

```
// Answer to the question no: 03 //

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

const int N=100;
int dp[N][N];

int main()
{
   string A,B;cin >> A >> B;
   for(int i=0; i<=A.size(); i++){
      for(int j=0; j<=B.size(); j++){
         if (i == 0 || j == 0){
            dp[i][j] = 0;
         }
         else{
            if(A[i-1] == B[j-1]){
               dp[i][j]= 1+ dp[i-1][j-1];
            }
            else{
               dp[i][j]=max(dp[i-1][j], dp[i][j-1]);
            }
         }
         //cout<<dp[i][j]<<' ';
      }
      cout<<'\n';
   }
   cout<<endl;
   cout<<"MAX ANS: "<<dp[A.size()][B.size()]<<'\n';
   return 0;
}
```