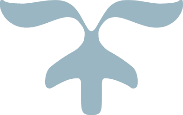


DAA WEEK – 12 SKILL – 12



# [Knapsack](https://www.hackerrank.com/contests/daa-skill-12-dynamic-programming-part-iii/challenges/unbounded-knapsack)

#include <stdio.h>

#include <stdbool.h>

void print(bool sack[], int k) {

for (int i = k; i >= 0; i--) {

if (sack[i]) {

printf("%d\n", i);

return;

}

}

printf("0\n");

}

int main() {

int t;

scanf("%d", &t);

while (t--) {

int n, k;

scanf("%d %d", &n, &k);

int weights[n];

bool sack[k + 1];

for (int i = 0; i <= k; i++)

sack[i] = false;

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

scanf("%d", &weights[i]);

if (weights[i] < k + 1)

sack[weights[i]] = true;

}

for (int i = 0; i <= k; i++) {

for (int j = 0; j < n; j++) {

if (sack[i]) break;

if (i - weights[j] > 0 && sack[i - weights[j]])

sack[i] = true;

}

}

print(sack, k);

}

return 0;

}

**Knapsack Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

# [Bricks Game](https://www.hackerrank.com/contests/daa-skill-12-dynamic-programming-part-iii/challenges/play-game)

#include <stdio.h>

#include <stdlib.h>

int main() {

int T;

scanf("%d", &T);

while (T > 0) {

int N;

scanf("%d", &N);

long long \*bricks = (long long \*)malloc(N \* sizeof(long long));

long long \*sum = (long long \*)malloc(N \* sizeof(long long));

for (int i = 0; i < N; i++) {

scanf("%lld", &bricks[N - i - 1]);

}

sum[0] = bricks[0];

for (int i = 1; i < N; i++) {

sum[i] = sum[i - 1] + bricks[i];

}

if (N < 4) {

printf("%lld\n", sum[N - 1]);

T--;

} else {

long long \*dp = (long long \*)malloc(N \* sizeof(long long));

dp[0] = sum[0];

dp[1] = sum[1];

dp[2] = sum[2];

for (int i = 3; i < N; i++) {

dp[i] = sum[i] - dp[i - 3];

if (dp[i] < sum[i] - dp[i - 2]) dp[i] = sum[i] - dp[i - 2];

if (dp[i] < sum[i] - dp[i - 1]) dp[i] = sum[i] - dp[i - 1];

}

printf("%lld\n", dp[N - 1]);

free(dp);

T--;

}

free(bricks);

free(sum);

}

return 0;

}

**Bricks Game Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

# [Coin on the Table](https://www.hackerrank.com/contests/daa-skill-12-dynamic-programming-part-iii/challenges/coin-on-the-table)

#include <stdio.h>

#include <string.h>

char a[52][52];

int b[52][52][1001];

int n, m, k;

void dfs(int x, int y, int t, int v) {

if (x < 0 || x >= n || y < 0 || y >= m || t > k) return;

if (v < b[x][y][t] || b[x][y][t] == -1) {

b[x][y][t] = v;

if (a[x][y] != '\*') {

dfs(x, y + 1, t + 1, v + (a[x][y] != 'R'));

dfs(x, y - 1, t + 1, v + (a[x][y] != 'L'));

dfs(x - 1, y, t + 1, v + (a[x][y] != 'U'));

dfs(x + 1, y, t + 1, v + (a[x][y] != 'D'));

} else {

dfs(x, y, t + 1, v);

}

}

}

int main() {

scanf("%d %d %d", &n, &m, &k);

int sx = -1, sy = -1;

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

for (int j = 0; j < m; j++) {

scanf(" %c", &a[i][j]);

if (a[i][j] == '\*') {

sx = i;

sy = j;

}

}

}

memset(b, -1, sizeof(b));

dfs(0, 0, 0, 0);

if (sx == -1 || sy == -1) {

printf("-1\n");

} else {

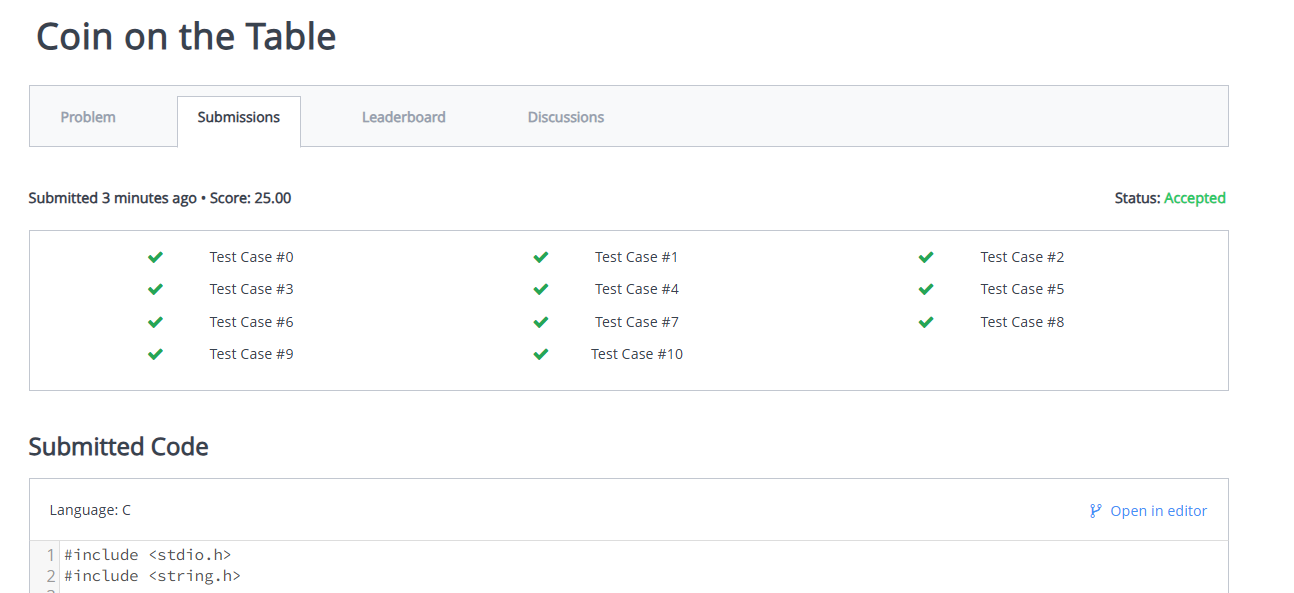
printf("%d\n", b[sx][sy][k]);

}

return 0;

}

**Coin on the Table Test Cases**

****

# [Play with words](https://www.hackerrank.com/contests/daa-skill-12-dynamic-programming-part-iii/challenges/strplay)

#include <stdio.h>

#include <string.h>

int L[3100][3100];

int max(int a, int b) {

return (a > b) ? a : b;

}

int main() {

char str[4000];

int i, j, l, cl;

scanf("%s", str);

l = strlen(str);

for (i = 0; i < l; i++) {

L[i][i] = 1;

}

for (cl = 2; cl <= l; cl++) {

for (i = 0; i < l - cl + 1; i++) {

j = i + cl - 1;

if (str[i] == str[j] && cl == 2)

L[i][j] = 2;

else if (str[i] == str[j])

L[i][j] = L[i + 1][j - 1] + 2;

else

L[i][j] = max(L[i][j - 1], L[i + 1][j]);

}

}

int mx = 0;

for (i = 0; i < l - 1; i++) {

if (L[0][i] \* L[i + 1][l - 1] > mx) {

mx = L[0][i] \* L[i + 1][l - 1];

}

}

printf("%d\n", mx);

return 0;

}

**Play with words Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

**SKILL WEEK – 12**

<https://www.hackerrank.com/contests/daa-skill-12-dynamic-programming-part-iii/challenges>