OTH Struct and Stack

Nama: Jehova Putra Yan Nehru

Kelas: IF03-01

Nim: 1203230107

1. SOURCE CODE:

```
#include <stdio.h>

struct Node {
    struct Node* link;
    char alphabet;
};

int main() {
    struct Node l1, l2, l3, l4, l5, l6, l7, l8, l9;
    l1.link = NULL;
    l1.alphabet = 'F';

    l2.link = NULL;
    l2.alphabet = 'M';

    l3.link = NULL;
    l3.alphabet = 'A';

    l4.link = NULL;
    l4.alphabet = 'I';

    l5.link = NULL;
    l5.alphabet = 'K';

    l6.link = NULL;
    l6.alphabet = 'T';

    l7.link = NULL;
    l7.alphabet = 'N';

    l8.link = NULL;
    l8.alphabet = 'O';

    l9.link = NULL;
    l9.alphabet = 'R';

```

```
38.     l7.link = &l1; //N -> F
39.     l1.link = &l8; //F -> O
40.     l8.link = &l2; //O -> M
41.     l2.link = &l5; //M -> K
42.     l5.link = &l3; //K -> A
43.     l3.link = &l6; //A -> T
44.     l6.link = &l9; //T -> R
45.     l9.link = &l4; //R -> I
46.     l4.link = &l7; //I -> N
47.
48.     char word[] = {
49.         l3.link->link->link->alphabet,
50.         l3.link->link->link->link->alphabet,
51.         l3.link->link->link->link->link->alphabet,
52.         l3.link->link->link->link->link->link->alphabet,
53.         l3.link->link->alphabet,
54.         l3.link->link->link->link->link->link->link->alphabet,
55.         l3.link->link->link->link->link->link->link->link->link-
   >alphabet,
56.         l3.link->link->link->link->link->link->link->link->link->link-
   >alphabet,
57.         l3.link->link->link->alphabet,
58.         l3.link->link->link->link->link->link->link->link->alphabet,
59.         l3.link->link->link->link->link->link->link->link->link->link-
   >alphabet
60.     };
61.
62.     printf("%s", word);
63.
64.     return 0;
65. }
```

SS OUTPUT:

```
PS C:\Users\JEHOVA\Documents\VSC\Bahasa C or C++> cd "c:\Users\JEHOVA\Documents\VSC
INFORMATIKA
PS C:\Users\JEHOVA\Documents\VSC\Bahasa C or C++>
```

Penjelasan:

a. **Struct Node** digunakan untuk mengatur link dan alphabet
b. **Int Main** digunakan untuk mengisi struct node 11 – 19, juga untuk penginisialisasian alphabet, dan link dengan value null
c. **Char word** digunakan untuk mengisi array string yg digunakan untuk menyimpan kata yg akan dibentuk

2. SOURCE CODE:

```
#include <stdio.h>

int main() {
   long long int c, i, g, m, n, x, la, lb;
   long long int a[100010], b[100010];
   scanf("%lld", &g);
   while (g--) {
      la = lb = 0;
      scanf("%lld%lld%lld", &n, &m, &x);
      scanf("%lld", &a[1]);
      for (i = 1; ++i <= n;) {
         scanf("%lld", &a[i]);
         a[i] += a[i - 1];
      }
      scanf("%lld", &b[1]);
      for (i = 1; ++i <= m;) {
         scanf("%lld", &b[i]);
         b[i] += b[i - 1];
      }
      la = 1;
      while (la <= n && a[la] <= x)
         la++;
      la--;
      c = la;
      lb = 1;
      while (lb <= m && b[lb] <= x) {
         if (la && b[lb] + a[la] > x)
            la--;
         else {
            if (c < la + lb)
               c = la + lb;
            lb++;
         }
      }
      printf("%lld\n", c);
   }
   return 0;
}
```

SS OUTPUT:


Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
Input (stdin)
Download
1
5 4 10
4 2 4 6 1
2 1 8 5
Your Output (stdout)
4
Expected Output
Download
4