Nama : M Faruq Rantisi

Nim : 1203230122

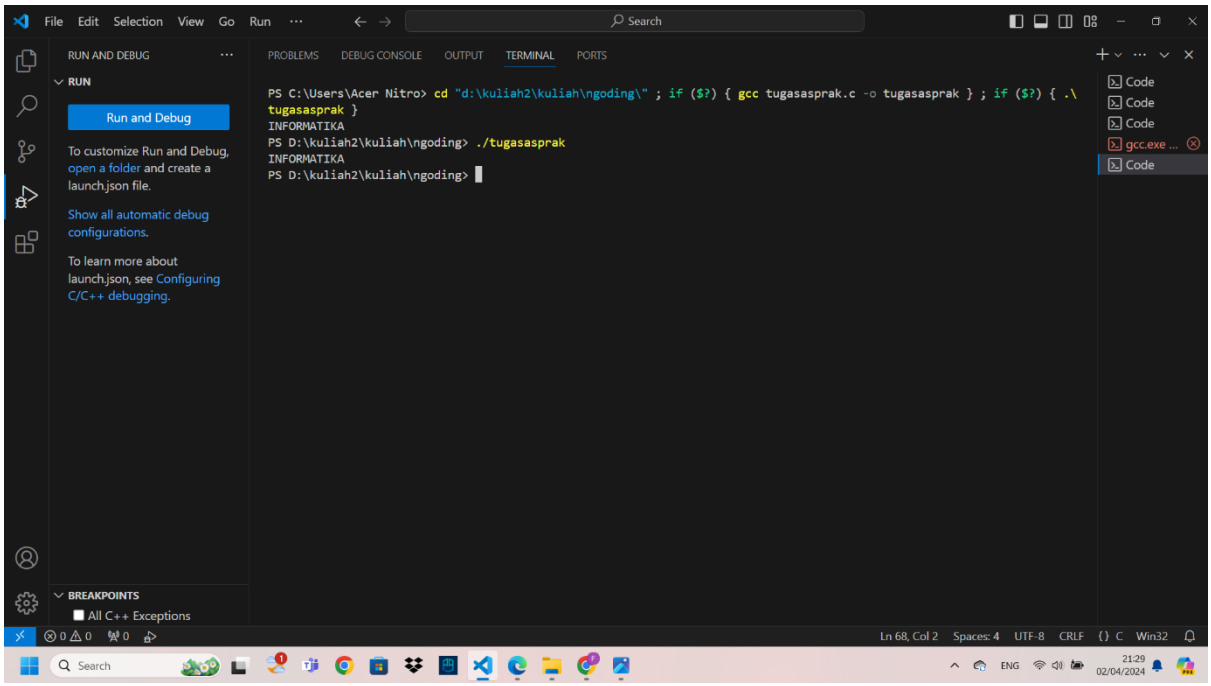
TUGAS 1

Source code :

```c
#include <stdio.h>


// Mendefinisikan struct Batu untuk menyimpan HURUF
struct Batu {
    char HURUF;  // Menyimpan huruf pada batu
    struct Batu *link;  // Pointer ke Batu berikutnya dalam urutan
};

int main() {
    // Inisialisasi batu
    struct Batu l1, l2, l3, l4, l5, l6, l7, l8, l9;

    l1.link = NULL;
    l1.HURUF = 'F';

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

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

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

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

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

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

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

    l9.link = NULL;
    l9.HURUF = 'R';
```

```c
    // Mengatur koneksi
    l7.link = &l1;
    l1.link = &l8;
    l8.link = &l2;
    l2.link = &l5;
    l5.link = &l3;
    l3.link = &l6;
    l6.link = &l9;
    l9.link = &l4;
    l4.link = &l7;


    // Mengakses huruf pada batu menggunakan l3 sebagai titik awal
    printf("%c", l3.link->link->link->HURUF);
    printf("%c", l3.link->link->link->link->HURUF);
    printf("%c", l3.link->link->link->link->link->HURUF);
    printf("%c", l3.link->link->link->link->link->link->HURUF);
    printf("%c", l3.link->link->HURUF);
    printf("%c", l3.link->link->link->link->link->link->link->HURUF);
    printf("%c", l3.HURUF);
    printf("%c", l3.link->HURUF);
    printf("%c", l3.link->link->link->HURUF);
    printf("%c", l3.link->link->link->link->link->link->link->link->HURUF);
    printf("%c", l3.HURUF);

    return 0;
}
```

SS output :



Penjelasan code :

1. #include <stdio.h> : Mendefinisikan bahwa kita akan menggunakan fungsi-fungsi standar input-output dari library stdio.h.

2. struct Batu { // Mendefinisikan struct Batu untuk menyimpan HURUF
3. char HURUF; // Menyimpan huruf pada batu
4. struct Batu *link; // Pointer ke Batu berikutnya dalam urutan

5. int main() { //fungsi utama
6. // Inisialisasi batu
7. struct Batu l1, l2, l3, l4, l5, l6, l7, l8, l9; // Membuat instansi dari Batu yaitu l1, l2, dst.

8. l1.link = NULL; // Mengatur link l1 menjadi NULL
9. l1.HURUF = 'F'; // Mengatur HURUF l1 menjadi 'F'

10. l2.link = NULL; // Mengatur link l2 menjadi NULL
11. l2.HURUF = 'M'; // Mengatur HURUF l2 menjadi 'M'

12. l3.link = NULL; // Mengatur link l3 menjadi NULL
13. l3.HURUF = 'A'; // Mengatur HURUF l3 menjadi 'A'

14. l4.link = NULL; // Mengatur link l4 menjadi NULL
15. l4.HURUF = 'I'; // Mengatur HURUF l4 menjadi 'I'

16. l5.link = NULL; // Mengatur link l5 menjadi NULL
17. l5.HURUF = 'K'; // Mengatur HURUF l5 menjadi 'K'

18. l6.link = NULL; // Mengatur link l6 menjadi NULL
19. l6.HURUF = 'T'; // Mengatur HURUF l6 menjadi 'T'

20. l7.link = NULL; // Mengatur link l7 menjadi NULL
21. l7.HURUF = 'N'; // Mengatur HURUF l7 menjadi 'N'

22. l8.link = NULL; // Mengatur link l8 menjadi NULL
23. l8.HURUF = 'O'; // Mengatur HURUF l8 menjadi 'O'

24. l9.link = NULL; // Mengatur link l9 menjadi NULL
25. l9.HURUF = 'R'; // Mengatur HURUF l9 menjadi 'R'

26. l7.link = &l1; // Mengatur link l7 menjadi alamat dari l1
27. l1.link = &l8; // Mengatur link l1 menjadi alamat dari l8
28. l8.link = &l2; // Mengatur link l8 menjadi alamat dari l2
29. l2.link = &l5; // Mengatur link l2 menjadi alamat dari l5
30. l5.link = &l3; // Mengatur link l5 menjadi alamat dari l3
31. l3.link = &l6; // Mengatur link l3 menjadi alamat dari l6
32. l6.link = &l9; // Mengatur link l6 menjadi alamat dari l9
33. l9.link = &l4; // Mengatur link l9 menjadi alamat dari l4
34. l4.link = &l7; // Mengatur link l4 menjadi alamat dari l7

35. printf("%c", l3.link->link->link->HURUF); // Mengakses huruf ke-3 dari l3
36. printf("%c", l3.link->link->link->link->HURUF); // Mengakses huruf ke-4 dari l3
37. printf("%c", l3.link->link->link->link->link->HURUF); // Mengakses huruf ke-5 dari l3
38. printf("%c", l3.link->link->link->link->link->link->HURUF); // Mengakses huruf ke-6 dari l3
39. printf("%c", l3.link->link->HURUF); // Mengakses huruf ke-2 dari l3
40. printf("%c", l3.link->link->link->link->link->link->link->HURUF); // Mengakses huruf ke-7 dari l3
41. printf("%c", l3.HURUF); // Mengakses huruf ke-1 dari l3
42. printf("%c", l3.link->HURUF); // Mengakses huruf ke-2 dari l3
43. printf("%c", l3.link->link->link->HURUF); // Mengakses huruf ke-4 dari l3
44. printf("%c", l3.link->link->link->link->link->link->link->link->HURUF); // Mengakses huruf ke-9 dari l3
45. printf("%c", l3.HURUF); // Mengakses huruf ke-1 dari l3
46. return 0; : program telah selesai

TUGAS 2

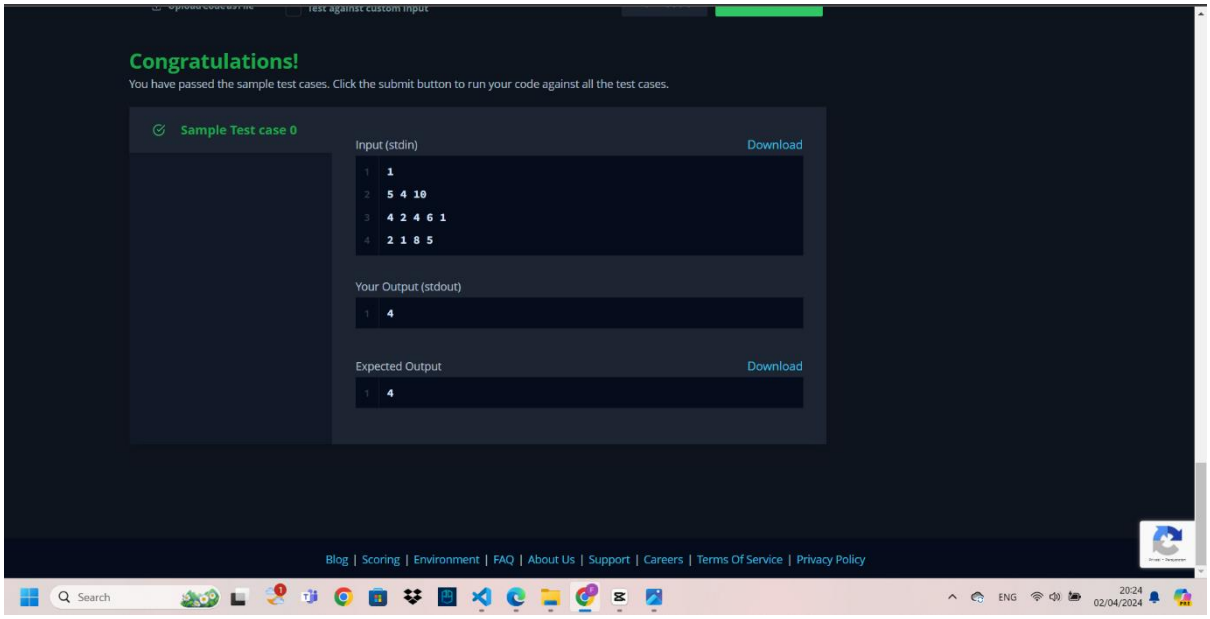
Source code :

```c
#include <stdio.h>

// Fungsi untuk mencari nilai maksimum dari dua bilangan
int max(int a, int b) {
    return a > b ? a : b;
}
```

```
// Fungsi untuk mencari jumlah maksimum elemen yang dapat diambil dari kedua
stack
int twoStacks(int maxSum, int a[], int m, int b[], int n) {
    int total = 0, count = 0, maxCount = 0;
    int i = 0, j = 0;

    // Iterasi untuk mengambil elemen dari kedua stack
    while (i < m && total + a[i] <= maxSum) {
        total += a[i++];
        count++;
        maxCount = max(maxCount, count);
    }

    // Iterasi sambil mempertimbangkan elemen dari stack kedua
    while (j < n && i >= 0) {
        total += b[j++];
        count++;

        // Jika total melebihi maxSum, kurangi elemen dari stack pertama
        while (total > maxSum && i > 0) {
            i--;
            total -= a[i];
            count--;
        }

        // Perbarui maxCount jika jumlah saat ini lebih besar
        if (total <= maxSum) {
            maxCount = max(maxCount, count);
        }
    }

    return maxCount;
}

int main() {
    int games;
    scanf("%d", &games);

    for (int g = 0; g < games; g++) {
        int m, n, maxSum;
        scanf("%d %d %d", &m, &n, &maxSum);
        int a[m], b[n];

        // Masukkan elemen stack pertama
        for (int i = 0; i < m; i++) {
            scanf("%d", &a[i]);
        }
```

```
        // Masukkan elemen stack kedua
        for (int i = 0; i < n; i++) {
            scanf("%d", &b[i]);
        }

        // Cetak hasil jumlah maksimum elemen yang dapat diambil
        printf("%d\n", twoStacks(maxSum, a, m, b, n));
    }

    return 0;

}
```

SS OUTPUT :



Penjelasan code :

1) #include <stdio.h> // Menggunakan fungsi-fungsi standar input-output dari library stdio.h

2) int max(int a, int b) {// Fungsi untuk mencari nilai maksimum dari dua bilangan
3) return a > b ? a : b; // Mengembalikan nilai maksimum dari a dan b

4) int twoStacks(int maxSum, int a[], int m, int b[], int n) {// Fungsi untuk mencari jumlah maksimum elemen yang dapat diambil dari kedua stack
5) int total = 0, count = 0, maxCount = 0;
6) int i = 0, j = 0;

7) while (i < m && total + a[i] <= maxSum) {// Iterasi untuk mengambil elemen dari kedua stack
8) total += a[i++];
9) count++;
10) maxCount = max(maxCount, count);

11) while (j < n && i >= 0) {// Iterasi sambil mempertimbangkan elemen dari stack kedua
12) total += b[j++];
13) count++;

14) while (total > maxSum && i > 0) {// Jika total melebihi maxSum, kurangi elemen dari stack pertama
    a) i--;
    b) total -= a[i];
    c) count--;

15) if (total <= maxSum) { // Perbarui maxCount jika jumlah saat ini lebih besar
16)
    a) maxCount = max(maxCount, count);

17) return maxCount; // Mengembalikan jumlah maksimum elemen yang dapat diambil dari kedua stack

18) int main() { //fungsi utama
19) int games;
20) scanf("%d", &games); // Meminta input jumlah games yang akan dimainkan

21) for (int g = 0; g < games; g++) { // Loop sejumlah games
22) int m, n, maxSum;
23) scanf("%d %d %d", &m, &n, &maxSum); // Meminta input jumlah elemen dalam stack pertama, stack kedua, dan maxSum
24) int a[m], b[n];
25) for (int i = 0; i < m; i++) {// Masukkan elemen stack pertama
    a) scanf("%d", &a[i]);

26) for (int i = 0; i < n; i++) {// Masukkan elemen stack kedua
    a) scanf("%d", &b[i]);

27) printf("%d\n", twoStacks(maxSum, a, m, b, n)); // Cetak hasil jumlah maksimum elemen yang dapat diambil

28) return 0; // Program selesai