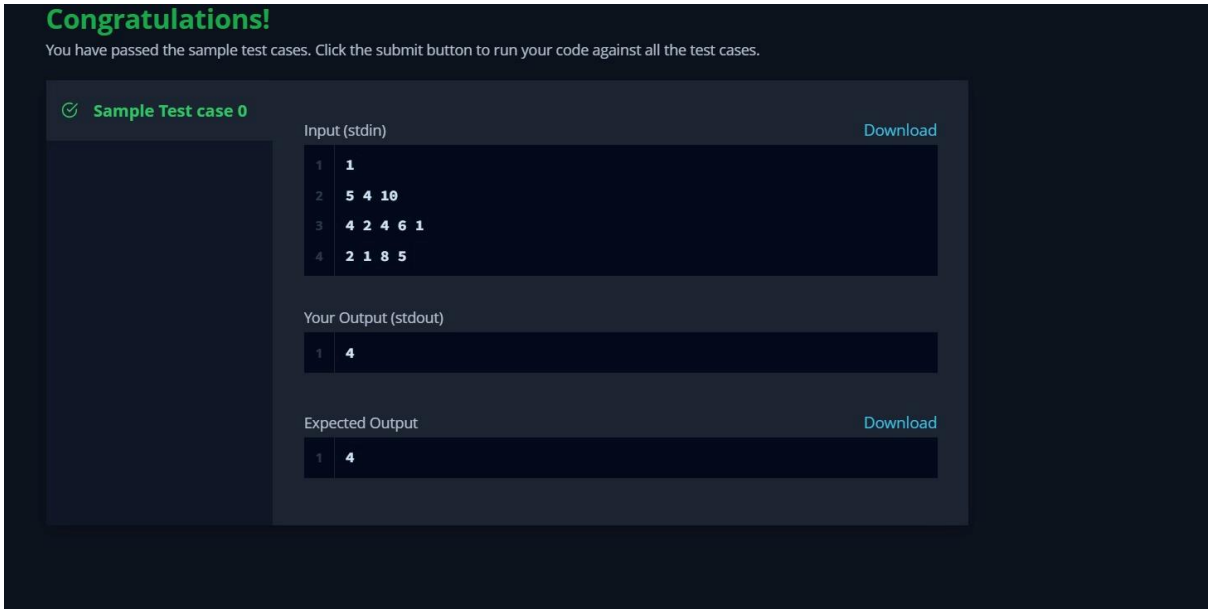
AZTECA CHELSY PRAMESTIA

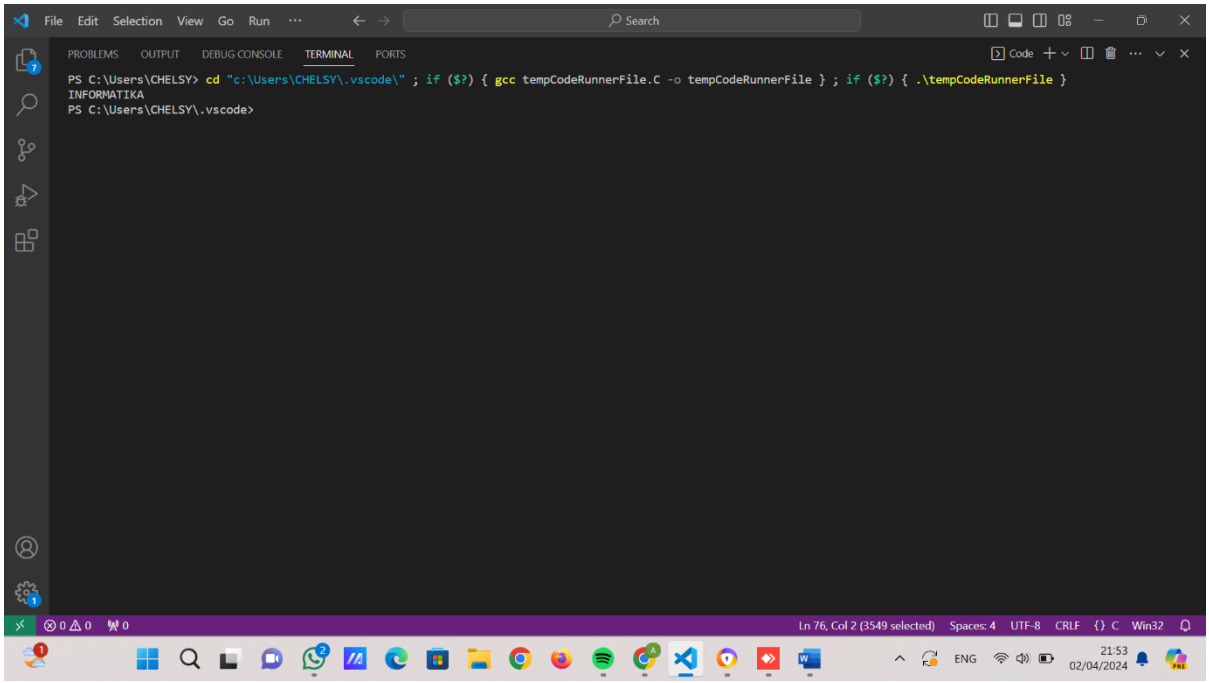
1203230062

OTH

1.

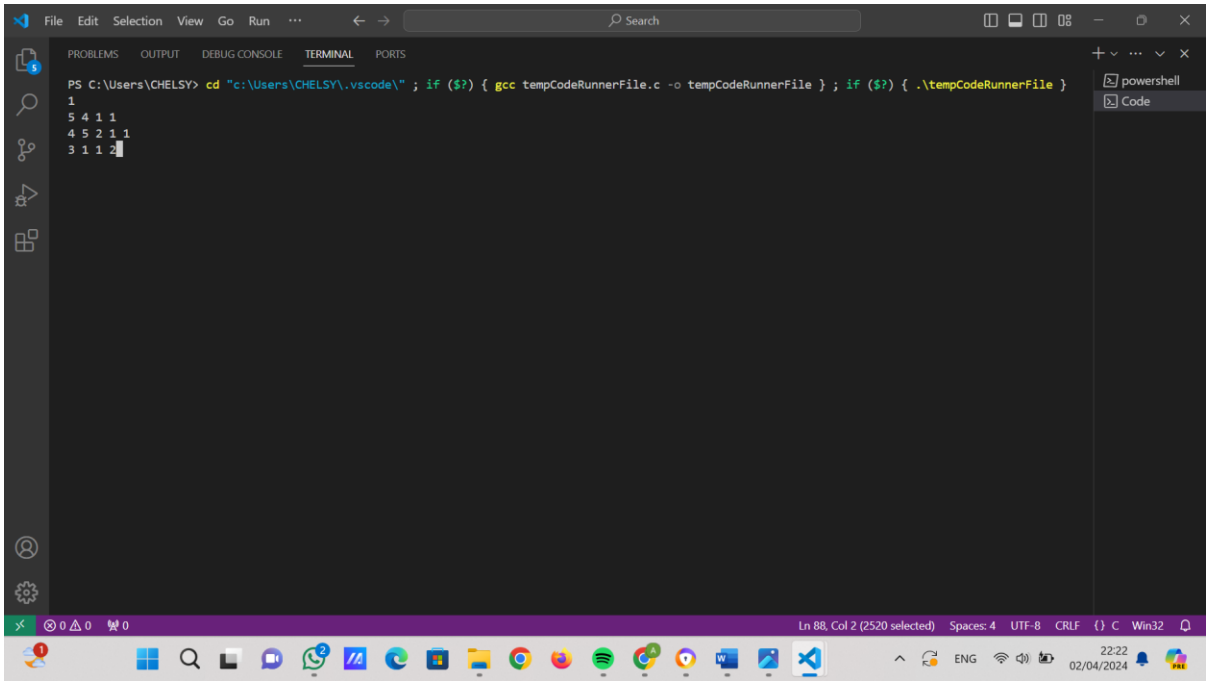


```c
#include <stdio.h>


// 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';
    //Sembilan variabel tipe struktur Batu diinisialisasi yaitu l1 hingga l9.
  //Setiap variabel dalam struktur Batu diinisialisasi dengan karakter  yang berbeda.
  //Misal l1 diinisialisasi dengan huruf yang mengandung huruf 'F', l2 diinisialisasi dengan huruf yang mengandung huruf 'M',
  //dan seterusnya hingga l9 diinisialisasi dengan huruf yang mengandung huruf 'R'.

  //Semua variabel tautan disetel ke NULL, yang menunjukkan bahwa tidak ada tautan ke struktur Batu lainnya.




    // Mengatur koneksi
    l7.link = &l1;// Pointer link dari l7 diatur untuk menunjuk ke l1, sehingga l7 menunjuk ke l1.
    l1.link = &l8;// Pointer link dari l1 diatur untuk menunjuk ke l8, sehingga l1 menunjuk ke l8.
    l8.link = &l2;//Pointer link dari l8 diatur untuk menunjuk ke l2, sehingga l8 menunjuk ke l2.
    l2.link = &l5;// Pointer link dari l2 diatur untuk menunjuk ke l5, sehingga l2 menunjuk ke l5.
    l5.link = &l3;//Pointer link dari l5 diatur untuk menunjuk ke l3, sehingga l5 menunjuk ke l3.
    l3.link = &l6;//Pointer link dari l3 diatur untuk menunjuk ke l6, sehingga l3 menunjuk ke l6.
    l6.link = &l9;//Pointer link dari l6 diatur untuk menunjuk ke l9, sehingga l6 menunjuk ke l9.
    l9.link = &l4;// Pointer link dari l9 diatur untuk menunjuk ke l4, sehingga l9 menunjuk ke l4.
    l4.link = &l7;// Pointer link dari l4 diatur untuk menunjuk ke l7, sehingga l4 menunjuk ke l7.


    // Mengakses huruf pada batu menggunakan l3 sebagai titik awal
    printf("%c", l3.link->link->link->HURUF);//mencetak karakter HURUF dari elemen yang terletak 3 langkah setelah l3.
    printf("%c", l3.link->link->link->link->HURUF);//mencetak karakter HURUF dari elemen yang terletak 4 langkah setelah l3.
    printf("%c", l3.link->link->link->link->link->HURUF);//mencetak karakter HURUF dari elemen yang terletak 5 langkah setelah l3.
    printf("%c", l3.link->link->link->link->link->link->HURUF);//mencetak karakter HURUF dari elemen yang terletak 6 langkah setelah l3.
    printf("%c", l3.link->link->HURUF);//mencetak karakter HURUF dari elemen yang terletak 2 langkah setelah l3.
    printf("%c", l3.link->link->link->link->link->link->link->HURUF);//mencetak karakter HURUF dari elemen yang terletak 7 langkah setelah l3.
    printf("%c", l3.HURUF); //mencetak karakter HURUF dari elemen l3 itu sendiri.
    printf("%c", l3.link->HURUF); //mencetak karakter HURUF dari elemen yang terletak 1 langkah setelah l3.
    printf("%c", l3.link->link->link->HURUF); //mencetak karakter HURUF dari elemen yang terletak 3 langkah setelah l3.
    printf("%c", l3.link->link->link->link->link->link->link->link->HURUF); //mencetak karakter HURUF dari elemen yang terletak 8 langkah setelah l3.
    printf("%c", l3.HURUF);//mencetak karakter HURUF dari elemen l3 itu sendiri.

    return 0;//program selesai
}
```


File Edit Selection View Go Run
Search
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code
PS C:\Users\CHELSY> cd "c:\Users\CHELSY\.vscode\" ; if ($?) { gcc tempCodeRunnerFile.C -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
INFORMATIKA
PS C:\Users\CHELSY\.vscode>
Ln 76, Col 2 (3549 selected) Spaces: 4 UTF-8 CRLF {} C Win32
ENG
21:53
02/04/2024

2.

```c
#include <stdio.h>
#include <stdlib.h>

// Fungsi untuk menampilkan isi tumpukan dalam bentuk visual
void cetakTumpukan(int* tumpukan, int atas) {// deklarasi fungsi cetakTumpukan yang menerima pointer ke tumpukan
//(yang diwakili sebagai array) dan indeks dari elemen teratas dalam tumpukan.
    printf("[");//mencetak []
    for (int i = 0; i <= atas; i++){
        //untuk melakukan iterasi melalui setiap elemen dalam tumpukan hingga indeks teratas (atas).
        printf("%d", tumpukan[i]);// mencetak nilai dari setiap elemen tumpukan menggunakan printf. %d digunakan untuk mencetak nilai integer.
        if (i != atas) printf(", ");//Setiap kali kecuali elemen teratas, fungsi mencetak koma dan spasi ", "
        // untuk memisahkan nilai-nilai dalam tumpukan.
    }
    printf("]\n");//mencetak karakter "]" sebagai penanda akhir dari tumpukan, dan juga mencetak karakter newline "\n" untuk pindah baris.
}

// Fungsi untuk menyelesaikan masalah Two Stacks
int duaTumpukan(int maxJumlah, int jml_a, int* a, int jml_b, int* b) {
    //deklarasi fungsi duaTumpukan yang mengembalikan nilai integer. Fungsi menerima lima parameter: maxJumlah, jml_a, a, jml_b, dan b.
    int i = 0, j = 0, jumlah_tumpukan = 0, hitung = 0;//i dan j adalah variabel yang akan digunakan sebagai indeks untuk mengakses elemen
    //dalam array a dan b secara berurutan.
    //umlah_tumpukan adalah variabel yang akan digunakan untuk melacak jumlah total elemen yang telah diambil dari kedua tumpukan.
    //menghitung jumlah elemen yang telah diambil dari tumpukan a dalam satu iterasi.
    int max_hitung = 0;//Variabel max_hitung diinisialisasi dengan nilai 0.

    printf("Langkah 1: Mengambil elemen dari tumpukan A hingga melebihi maxJumlah atau mencapai akhir:\n");//mencetak langkah pertama
    while (i < jml_a && jumlah_tumpukan + a[i] <= maxJumlah) {// mengecek apakah indeks i masih kurang dari jumlah elemen dalam tumpukan A
    //(jml_a). Ini memastikan bahwa kita tidak melampaui batas tumpukan A.
        jumlah_tumpukan += a[i];//memeriksa apakah menambahkan nilai elemen ke-i tumpukan A (a[i]) ke jumlah_tumpukan masih memenuhi
        //atau tidak melebihi jumlah maksimum yang diizinkan (maxJumlah).
        printf("Tumpukan A: ");//mencetak tumpukkan A
        cetakTumpukan(a, i); // Visualisasi tumpukan A
        printf("Tumpukan B: ");//mencetak tumpukkan B
        cetakTumpukan(b, j); // Visualisasi tumpukan B
        printf("Jumlah saat ini: %d\n", jumlah_tumpukan);//mencetak jumlah total elemen yang telah diambil dari kedua tumpukan saat ini (jumlah_tumpukan).
        i++;
        hitung++;
    }


    printf("Jumlah akhir setelah Langkah 1: %d\n", jumlah_tumpukan);//mencetak jumlah total elemen yang telah diambil dari tumpukan A setelah langkah 1 selesai.
    max_hitung = hitung;//mengatur nilai dari variabel max_hitung sama dengan nilai dari variabel hitung



    printf("\nLangkah 2: Menambahkan elemen dari tumpukan B dan menyesuaikan jumlah:\n");//mencetak langgak kedua
    while (j < jml_b && i >= 0) {// mengecek apakah indeks j masih kurang dari jumlah elemen dalam tumpukan B (jml_b)
    //mengecek apakah indeks i masih lebih besar atau sama dengan 0.
        jumlah_tumpukan += b[j];//pengambilan elemen dari tumpukan B dan menambahkannya ke jumlah total elemen yang telah diambil dari
        //kedua tumpukan.
        printf("Tumpukan A: ");
        cetakTumpukan(a, i); // Visualisasi tumpukan A
        printf("Tumpukan B: ");
        cetakTumpukan(b, j); // Visualisasi tumpukan B
        printf("Jumlah saat ini: %d\n", jumlah_tumpukan);//mencetak jumlah total elemen yang telah diambil dari kedua tumpukan saat ini
        j++;

        while (i > 0 && jumlah_tumpukan > maxJumlah) {//mengecek apakah i lebih besar dari no1 dan jumlah_tumpukkan lebih besar dari
        //maxJumlah
            i--;//mengurangi nilai indeks i, sehingga algoritma akan mundur ke elemen sebelumnya dalam tumpukan A.
            jumlah_tumpukan -= a[i];//langkah yang mengurangi nilai dari jumlah total elemen yang telah diambil dari kedua tumpukan
            //(jumlah_tumpukan) dengan nilai elemen yang dihapus dari tumpukan A (a[i])
            hitung--;//mengurangi nilai dari variabel hitung
        }

        if (jumlah_tumpukan <= maxJumlah) {//mengecek apakah jumlah tumpukkan kurang dari ssamadengan max jumlah
            max_hitung = max_hitung > (hitung + j) ? max_hitung : (hitung + j);
            //Jika nilai dari max_hitung lebih besar dari hitung + j, maka nilai max_hitung tidak berubah.
            //jika nilai dari max_hitung tidak lebih besar dari hitung + j, maka nilai max_hitung akan diubah menjadi hitung + j
        }
    }

    return max_hitung;// nilai max_hitung akan dikembalikan ke bagian yang memanggil fungsi duaTumpukan
}

int main() {//fungsi utama
    int g;//variabel g tipe data integer
    scanf("%d", &g);//membaca input dari pengguna berupa bilangan bulat dan menyimpanya di variabel g

    for (int g_itr = 0; g_itr < g; g_itr++) {//menginisialisasi variabel g_itr dengan nilai 0, dan kemudian loop akan dieksekusi
    // selama g_itr kurang dari g. Setiap kali loop berjalan, g_itr akan bertambah satu.
        int n, m, maxJumlah;// Variabel-variabel ini akan digunakan untuk menyimpan nilai input yang akan dibaca.
        scanf("%d %d %d", &n, &m, &maxJumlah);//membaca tiga nilai integer dari input pengguna

        int* a = malloc(n * sizeof(int));//digunakan untuk mengalokasikan memori sejumlah n * sizeof(int)
        for (int i = 0; i < n; i++) {//perulangan yang dimulai dari i=0 dan akan terus berjalan selama i < n
            scanf("%d", &a[i]);//e fungsi scanf yang membaca nilai integer dari input pengguna dan menyimpannya dalam array a
        }

        int* b = malloc(m * sizeof(int));///digunakan untuk mengalokasikan memori sejumlah m * sizeof(int)
        for (int i = 0; i < m; i++) {//perulangan yang dimulai dari i=0 dan akan terus berjalan i<m
            scanf("%d", &b[i]);//fungsi scanf yang membaca nilai integer dari input pengguna dan menyimpannya dalam array b
        }

        printf("\nKasus Uji %d:\n", g_itr + 1);
        //mencetak informasi tentang kasus uji yang sedang diproses. g_itr + 1 digunakan untuk mencetak nomor kasus uji secara
        //berturut-turut, dimulai dari 1, karena indeks array dimulai dari 0.
        int hasil = duaTumpukan(maxJumlah, n, a, m, b);//arameter-parameter yang diperlukan oleh fungsi duaTumpukan.
        printf("\nMaksimum elemen yang dapat diambil: %d\n", hasil);//mencetak hasil dari kasus uji yang sedang diproses.
        //Nilai dari variabel hasil dicetak


        free(a);
        free(b);
        //digunakan untuk membebaskan memori yang dialokasikan untuk array a dan b
    }

    return 0;
}
```


File Edit Selection View Go Run ···
Search
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\CHELSY> cd "c:\Users\CHELSY\.vscode\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
1
5 4 1 1
4 5 2 1 1
3 1 1 2
powershell
Code
Ln 88, Col 2 (2520 selected) Spaces: 4 UTF-8 CRLF {} C Win32
ENG
22:22
02/04/2024