

NAMA : BENEDICTA EKA WIJAYANTI

KELAS : IF 03 02

NIM : 1203230059

SOAL NOMOR 1

1. SOURCE CODE

```
// soal nomor 1 //

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

struct Node
{
    char *abjad;
    struct Node *next;
};

int main()
{
    // deklarasi node
    struct Node l1, l2, l3, l4, l5, l6, l7, l8, l9;
    struct Node *next;

    // inisialisasi node
    l1.next = NULL;
    l1.abjad = "F";

    l2.next = NULL;
    l2.abjad = "M";

    l3.next = NULL;
    l3.abjad = "A";

    l4.next = NULL;
    l4.abjad = "I";

    l5.next = NULL;
    l5.abjad = "K";

    l6.next = NULL;
    l6.abjad = "T";

    l7.next = NULL;
    l7.abjad = "N";
```

```

18.next      = NULL;
18.abjad     = "O";

19.next      = NULL;
19.abjad     = "R";

// menghubungkan node sesuai dengan urutan
17.next = &l1;
11.next = &l8;
18.next = &l2;
12.next = &l5;
15.next = &l3;
13.next = &l6;
16.next = &l9;
19.next = &l4;
14.next = &l7;

// printf
printf("%s", 13.next->next->next->abjad); //
menampilkan huruf I
printf("%s", 13.next->next->next->next->abjad); //
menampilkan huruf N
printf("%s", 13.next->next->next->next->next->abjad); //
menampilkan huruf F
printf("%s", 13.next->next->next->next->next->next->abjad); //
menampilkan abjad O
printf("%s", 13.next->next->abjad); //
menampilkan abjad R
printf("%s", 13.next->next->next->next->next->next->next->abjad); //
menampilkan abjad M
printf("%s", 13.abjad); //
menampilkan abjad A
printf("%s", 13.next->abjad); //
menampilkan abjad T
printf("%s", 13.next->next->next->abjad); //
menampilkan abjad I
printf("%s", 13.next->next->next->next->next->next->next->next->abjad); //
menampilkan abjad K
printf("%s", 13.abjad); //
menampilkan abjad A
return 0;
}

```

2. PENJELASAN

```

// membuat struktur data bernama Node
struct Node

```

```
{
    char *abjad;           // pointer untuk menyimpan huruf alphabet (string)
    struct Node *next;      // pointer untuk mengarahkan ke Node lain
};
```

```
int main()
{
    // deklarasi node
    struct Node l1, l2, l3, l4, l5, l6, l7, l8, l9;    // bertipe data struct
Node
    struct Node *next;
```

```
// inisialisasi node
    l1.next      = NULL;           //next di set ke NULL untuk menandakan belum
terhubung ke node lain
    l1.abjad     = "F";           // menginisialisasi l1 dengan huruf F (artinya
l1 itu hurufnya F)

    l2.next      = NULL;           //next di set ke NULL untuk menandakan belum
terhubung ke node lain
    l2.abjad     = "M";           // huruf l2 M

    l3.next      = NULL;           //next di set ke NULL untuk menandakan belum
terhubung ke node lain
    l3.abjad     = "A";           // huruf l3 A

    l4.next      = NULL;           //next di set ke NULL untuk menandakan belum
terhubung ke node lain
    l4.abjad     = "I";

    l5.next      = NULL;           //next di set ke NULL untuk menandakan belum
terhubung ke node lain
    l5.abjad     = "K";

    l6.next      = NULL;           //next di set ke NULL untuk menandakan belum
terhubung ke node lain
    l6.abjad     = "T";

    l7.next      = NULL;           //next di set ke NULL untuk menandakan belum
terhubung ke node lain
    l7.abjad     = "N";

    l8.next      = NULL;           //next di set ke NULL untuk menandakan belum
terhubung ke node lain
    l8.abjad     = "O";
```

```
    19.next      = NULL;          //next di set ke NULL untuk menandakan belum
    terhubung ke node lain
    19.abjad     = "R";
```

```
// menghubungkan node sesuai dengan urutan
    17.next = &l1;          // menandakan node l7 terhubung ke node l1
    11.next = &l8;
    18.next = &l2;
    12.next = &l5;
    15.next = &l3;
    13.next = &l6;
    16.next = &l9;
    19.next = &l4;
    14.next = &l7;          // l4 dihubungkan dengan l7 agar membentuk loop
    tertutup
```

```
// mencetak abjad yang dimulai dari l3
    //l3.next = l3 dihubungkan sm l6, artinya huruf dil3 itu T. supaya dpt I
    harus di next 2 kali
    printf("%s", l3.next->next->next->abjad);          //
    menampilkan huruf I
    printf("%s", l3.next->next->next->next->abjad);      //
    menampilkan huruf N
    printf("%s", l3.next->next->next->next->next->abjad); //
    menampilkan huruf F
    printf("%s", l3.next->next->next->next->next->next->abjad); //
    menampilkan abjad O
    printf("%s", l3.next->next->abjad);                  //
    menampilkan abjad R
    printf("%s", l3.next->next->next->next->next->next->next->abjad); //
    menampilkan abjad M

    // disini huruf dil3 kalau ga di next kan tetep A. jadi langsung dicetak
    pakai abjad
    printf("%s", l3.abjad);                             //
    menampilkan abjad A
    printf("%s", l3.next->abjad);                         //
    menampilkan abjad T
    printf("%s", l3.next->next->next->abjad);              //
    menampilkan abjad I
    printf("%s", l3.next->next->next->next->next->next->next->next->abjad); //
    menampilkan abjad K
    printf("%s", l3.abjad);                             //
    menampilkan abjad A
    return 0;
    // direturn 0 artinya program selesai berjalan
}
```

3. OUTPUT

```
PS C:\C> cd "c:\C\" ; if ($?) { gcc prak_asd_structdanstack.c -o prak_asd_structdanstack } ; if ($?) { .\prak_asd_structdanstack }  
INFORMATIKA
```

SOAL NOMOR 2

1. SOURCE CODE

```
// soal nomor 2 //

#include <stdio.h>

int twoStacks(int penjumlahanMaks, int arr1[], int size_arr1, int arr2[], int size_arr2)
{
    int penjumlahan = 0, count = 0, temp = 0, i = 0, j = 0;

    while (i < size_arr1 && penjumlahan + arr1[i] <= penjumlahanMaks)
    {
        penjumlahan += arr1[i++];
    }
    count = i;

    while (j < size_arr2 && i >= 0)
    {
        penjumlahan += arr2[j++];
        while (penjumlahan > penjumlahanMaks && i > 0)
        {
            penjumlahan -= arr1[--i];
        }
        if (penjumlahan <= penjumlahanMaks && i + j > count)
        {
            count = i + j;
        }
    }
    return count;
}

int main()
{
    int g;
    printf("Masukkan baris angka : \n");
    scanf("%d", &g);
    while (g-->0)
    {
        int n, m, penjumlahanMaks;
```

```

scanf("%d%d%d", &n, &m, &penjumlahanMaks);
int arr1[n], arr2[m];
for (int i = 0; i < n; i++)
{
    scanf("%d", &arr1[i]);
}
for (int i = 0; i < m; i++)
{
    scanf("%d", &arr2[i]);
}
printf("output : %d\n", twoStacks(penjumlahanMaks, arr1, n, arr2, m));
}
return 0;
}

```

2. PENJELASAN

```

// fungsi twoStacks dengan 5 parameter -> akan mencari kombinasi angka
// terbesar dari arr1 dan arr2 yang tidak melebihi penjumlahanMaks
int twoStacks(int penjumlahanMaks, int arr1[], int size_arr1, int arr2[], int
size_arr2)
// penjumlahanMaks -> untuk batas maksimum penjumlahan angka
// arr1 -> adalah array pertama yang berisi angka
// size_arr1 -> ukuran untuk arr1
// arr2 -> adalah array kedua yang berisi angka
// size_arr2 -> ukuran untuk arr2
{

```

```

int penjumlahan = 0, count = 0, temp = 0, i = 0, j = 0;
// penjumlahan -> adalah variabel utk menyimpan penjumlahan angka
// sementara yang diinisialisasi dengan 0
// count -> untuk menyimpan jumlah angka dalam kombinasi terbesar
// yang ditemukan
// temp -> tempat penyimpanan sementara
// i -> untuk iterasi arr1 dan j -> untuk iterasi arr2

```

```

// looping pertama untuk arr1.
// loop akan terus berjalan selama i kurang dari size_arr1 dan penjumlahan
// tidak melebihi batas penjumlahan maks (loop akan berjalan ketika ke 2
// kondisi terpenuhi)
while (i < size_arr1 && penjumlahan + arr1[i] <= penjumlahanMaks)
{
    // saat looping, nilai arr1 pada indeks ke-i akan ditambahkan kedalam
    // penjumlahan dan nilai i akan bertambah 1.
    penjumlahan += arr1[i++];
}

```

```
// ketika loop selesai, nilai i akan disimpan kedalam count  
count = i;
```

```
// contoh : arr1 = {1,2,3,4,5}  
// size_arr 1 = 5 dan penjumlahanMaks = 10  
// loop akan dimulai dengan i = 0, karena 0<5 dan 0+1 <= 10 maka while  
akan dijalankan  
// didalam loop, penjumlahan berubah jadi 1 (karena 0+1 = 1)  
// i nya di increment jadi 1  
// 1<5 dan 1+2 <= 10, looping akan berjalan dan penjumlahan akan bernilai  
3 ( 1+2 = 3 ) dan i diincrement menjadi 2  
// 2<5 dan 3+3 <= 10, hasil penjumlahannya adalah 6 (3+3) dan i(3)  
// 3<5 dan 6+4 <= 10, hasil penjumlahan adalah 10 (6+4) dan i(4)  
// 4<5 maka loop akan berjalan lg, tetapi hasil penjumlahannya kan sdh 10.  
sedangkan di rumus 10 + 5 > 10 artinya salah. jd loop akan berhenti  
// lalu count akan diset dengan i, jadai count = 4
```

```
// ini adalah loop untuk arr2  
// kalau di arr2, iterasi untuk indeks nya adalah j  
// loop ini akan berjalan selama j bernilai kurang dari size_arr2 dan i >=  
0  
while (j < size_arr2 && i >= 0)  
{  
    // pada setiap iterasi, nilai arr2 pada indeks j akan ditambahkan  
kedalam penjumlahan dan j akan diincrement  
    penjumlahan += arr2[j++];  
  
    // nested loop atau loop ke 2  
    // loop ini berjalan selama penjumlahan tidak lebih besar dari  
penjumlahanMaks dan i tidak lebih besar dari 0  
    while (penjumlahan > penjumlahanMaks && i > 0)  
    {  
        // disini tiap iterasi dikurangi sm penjumlahan dan i akan  
dikurangi pd setiap iterasi  
        penjumlahan -= arr1[--i];  
    }  
    // pengecekan kondisi saat loop ke 2 selesai  
    // jika nilai penjumlahan <= penjumlahanmaks dan  
    // hasil penjumlahan i + j lebih besar dari count  
    if (penjumlahan <= penjumlahanMaks && i + j > count)  
    {  
        // maka count akan diupdate isinya menjadi i+j  
        count = i + j;  
    }  
}  
// setelah itu akan mengembalikan nilai count saat semua loop selesai  
dijalankan
```

```
return count;
```

```
// contoh :  
    // arr1 = {1,2,3,4,5} dan arr2 = {6,7,8,9,10}  
    // size_arr1 = 5 dan size_arr2 = 5  
    // penjumlahanMaks = 15  
  
    // loop while nambahin arr2[0] = 6 ke penjumlahan, nilai penjumlahan skrg  
= 6  
    // penjumlahan = 6, dan penjumlahanMaks = 15 artinya nestedloop ga jalan  
    // arr2[1] = 7, penjumlahan = 13 nilai penjumlahan masih kurang dari  
penjumlahanMaks maka nestedloop ga jalan  
    // arr2[2] = 8. penjumlahan = 21 nilai penjumlahan lebih besar dari  
penjumlahanMaks, maka nestedloop berjalan  
    // nested loop ngurangin arr1[4] = 5 dari penjumlahan, jadi penjumlahan =  
16 dan i nya berubah menjadi 4 (karena di decrement)  
}
```

```
/ fungsi utama  
int main()  
{
```

```
// deklarasi g dengan tipe data integer  
    int g;  
    printf("Masukkan baris angka : \n"); // meminta user memasukkan baris  
angka  
    scanf("%d", &g); // baris angka yang disimpan  
kedalam g
```

```
// memulai loop while. dia jalan sebanyak nilai yang disimpan di g (nilai yg  
diinput user). pada tiap iterasi g di kurangi 1  
    while (g--)  
    {  
        // deklarasi variabel n m penjumlahanMaks dengan integer  
        int n, m, penjumlahanMaks;  
        // mengambil 3 integer dari user dan disimpan di n,m, penjumlahanMaks  
        scanf("%d%d%d", &n, &m, &penjumlahanMaks);  
  
        // deklarasi 2 array  
        // arr1 dengan size n, dan arr2 dengan size m  
        int arr1[n], arr2[m];  
  
        // loop pertama ngambil input untuk arr1  
        for (int i = 0; i < n; i++)  
        {  
            scanf("%d", &arr1[i]);
```



```

    }
    // loop untuk arr2
    for (int i = 0; i < m; i++)
    {
        scanf("%d", &arr2[i]);
    }

    // mencetak output kelayar dengan memanggil fungsi twoStacks dengan 4
parameter
    printf("output : %d\n", twoStacks(penjumlahanMaks, arr1, n, arr2, m));
}

// mengembalikan fungsi main
return 0;
}

```

3. OUTPUT

```

PS C:\C> cd "c:\C\" ; if ($?) { gcc prak_asd_structdanstack2.c -o p
rak_asd_structdanstack2 } ; if ($?) { .\prak_asd_structdanstack2 }
Masukkan baris angka :
1
5 4 10
4 2 4 6 1
2 1 8 5
output : 4
PS C:\C> 

```

```

PS C:\C> cd "c:\C\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCod
eRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan baris angka :
1
5 4 11
4 5 2 1 1
3 1 1 2
output : 5
PS C:\C> 

```