

Nama: I Gede Hermawan Adi Pranata
Nim: 1203230029
Kelas: IF-03-01

Komponen Penilaian	Ya	Tidak
Soal 1 sesuai dengan output yang diinginkan		
Soal 2 sesuai dengan output yang diinginkan		
Bonus soal 1 dikerjakan		

1. Refan sedang bermain judi kartu pada sebuah kasino, ia sedang memegang N kartu yang tidak terurut. Refan ingin mengurutkan kartu yang ia miliki dari terkecil hingga terbesar untuk mempermudah dalam permainan. Ia memindahkan posisi kartu dengan cara menukar dua posisi kartu yang ada di tangannya agar urut. Tentukanlah berapa minimal langkah pertukaran yang harus Refan lakukan untuk membuat kartu yang ia miliki menjadi urut. Program menerima 2 baris input, baris pertama berupa jumlah kartu dan baris kedua berupa angka atau nilai dari kartu yang dipisahkan dengan spasi. Nilai kartu yang dapat diinput adalah 1-10 dan J, Q, K. Dengan ketentuan $10 < J < Q$.

Code

```
#include <stdio.h>
#include <string.h>

void swap(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

void printCard(int number) {
    if (number == 1) printf("1 ");
    else if (number == 2) printf("2 ");
    else if (number == 3) printf("3 ");
    else if (number == 4) printf("4 ");
    else if (number == 5) printf("5 ");
    else if (number == 6) printf("6 ");
    else if (number == 7) printf("7 ");
    else if (number == 8) printf("8 ");
    else if (number == 9) printf("9 ");
    else if (number == 10) printf("10 ");
    else if (number == 11) printf("J ");
```

```

        else if (number == 12) printf("Q ");
        else if (number == 13) printf("K ");
        else printf("%d ", number);
    }

    void sortCards(int n, int *arr) {
        int swaps = 0;

        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    swap(&arr[j], &arr[j + 1]);
                    swaps++;

                    printf("Langkah %d: Pertukaran antara kartu %d (%c) dan kartu %d (%c)\n", swaps, j+1, arr[j], j+2, arr[j+1]);

                    printf("Kartu setelah pertukaran %d: ", swaps);
                    for (int k = 0; k < n; k++) {
                        printCard(arr[k]);
                    }
                    printf("\n");
                }
            }
        }

        printf("Jumlah pertukaran yang dilakukan : %d\n", swaps);
    }

    int main() {
        int n;
        printf("Masukkan kartu : ");
        scanf("%d", &n);

        int arr[n];
        char temp[3];
        printf("Masukkan angka kartu : ");
        for (int i = 0; i < n; i++) {
            scanf("%s", temp);
            if (strcmp(temp, "J") == 0) {
                arr[i] = 11;
            } else if (strcmp(temp, "Q") == 0) {
                arr[i] = 12;
            } else if (strcmp(temp, "K") == 0) {
                arr[i] = 13;
            }
        }
    }

```

	<pre> } else { sscanf(temp, "%d", &arr[i]); } } sortCards(n, arr); return 0; } </pre>
Penjelasan	<ul style="list-style-type: none"> • Library yang di-import: <ul style="list-style-type: none"> - <stdio.h>: Digunakan untuk fungsi input-output standar seperti printf() dan scanf(). - <string.h>: Digunakan untuk fungsi-fungsi terkait string seperti strcmp(). • Definisi Fungsi swap(): <ul style="list-style-type: none"> - Fungsi ini bertujuan untuk menukar dua nilai yang diberikan sebagai argumen menggunakan pointer. - int* a dan int* b adalah pointer ke dua variabel yang akan ditukar nilainya. - Dalam fungsi ini, nilai variabel a disimpan dalam variabel sementara temp, nilai variabel b disalin ke variabel a, dan nilai temp (yang berisi nilai awal a) disalin ke variabel b. • Definisi Fungsi printCard(): <ul style="list-style-type: none"> - Fungsi ini bertujuan untuk mencetak kartu sesuai dengan nilai numerik yang diberikan. - Pengecekan dilakukan untuk nilai numerik tertentu, seperti 11, 12, dan 13 yang akan dicetak sebagai "J", "Q", dan "K" masing-masing. - Untuk nilai numerik lainnya, langsung dicetak menggunakan %d. • Definisi Fungsi sortCards(): <ul style="list-style-type: none"> - Fungsi ini mengimplementasikan algoritma Bubble Sort untuk mengurutkan kartu-kartu yang diberikan. - Setiap kali ada pertukaran antara dua kartu, pesan yang berisi langkah pertukaran dan hasil kartu setelah pertukaran akan dicetak. - Jumlah pertukaran juga dicetak setelah proses pengurutan selesai. • Fungsi main(): <ul style="list-style-type: none"> - Pengguna diminta untuk memasukkan jumlah kartu dan nilai-nilai kartu tersebut. - Kartu-kartu dimasukkan ke dalam array arr, dengan penanganan khusus untuk kartu-kartu dengan nilai "J", "Q", dan "K". - Setelah itu, fungsi sortCards() dipanggil untuk mengurutkan kartu-kartu tersebut.

Output

```
PS D:\code\c tugas> cd "d:\code\c tugas\" ; if ($?) { gcc tugas.c
Masukkan kartu : 4
Masukkan angka kartu : 6 6 9 7
Langkah 1: Pertukaran antara kartu 3 ( ) dan kartu 4 ( )
Kartu setelah pertukaran 1: 6 6 7 9
Jumlah pertukaran yang dilakukan : 1
```

```
PS D:\code\c tugas> cd "d:\code\c tugas\" ; if ($?) { gcc tugas.c
Masukkan kartu : 5
Masukkan angka kartu : 3 2 8 7 4
Langkah 1: Pertukaran antara kartu 1 (♠) dan kartu 2 (♥)
Kartu setelah pertukaran 1: 2 3 8 7 4
Langkah 2: Pertukaran antara kartu 3 ( ) dan kartu 4 ( )
Kartu setelah pertukaran 2: 2 3 7 8 4
Langkah 3: Pertukaran antara kartu 4 (♦) dan kartu 5 ( )
Kartu setelah pertukaran 3: 2 3 7 4 8
Langkah 4: Pertukaran antara kartu 3 (♦) dan kartu 4 ( )
Kartu setelah pertukaran 4: 2 3 4 7 8
Jumlah pertukaran yang dilakukan : 4
```

```
PS D:\code\c tugas> cd "d:\code\c tugas\" ; if ($?) { gcc tugas.c
Masukkan kartu : 6
Masukkan angka kartu : 10 J K Q 3 2
Langkah 1: Pertukaran antara kartu 3 ( ) dan kartu 4 ( )
Kartu setelah pertukaran 1: 10 J Q K 3 2
Langkah 2: Pertukaran antara kartu 4 (♥) dan kartu 5 ( )
Kartu setelah pertukaran 2: 10 J Q 3 K 2
Langkah 3: Pertukaran antara kartu 5 (♠) dan kartu 6 ( )
Kartu setelah pertukaran 3: 10 J Q 3 2 K
Langkah 4: Pertukaran antara kartu 3 (♥) dan kartu 4 ( )
Kartu setelah pertukaran 4: 10 J 3 Q 2 K
Langkah 5: Pertukaran antara kartu 4 (♠) dan kartu 5 ( )
Kartu setelah pertukaran 5: 10 J 3 2 Q K
Langkah 6: Pertukaran antara kartu 2 (♥) dan kartu 3 ( )
Kartu setelah pertukaran 6: 10 3 J 2 Q K
Langkah 7: Pertukaran antara kartu 3 (♠) dan kartu 4 ( )
Kartu setelah pertukaran 7: 10 3 2 J Q K
Langkah 8: Pertukaran antara kartu 1 (♥) dan kartu 2 ( )
Kartu setelah pertukaran 8: 3 10 2 J Q K
Langkah 9: Pertukaran antara kartu 2 (♠) dan kartu 3 ( )
Kartu setelah pertukaran 9: 3 2 10 J Q K
Langkah 10: Pertukaran antara kartu 1 (♠) dan kartu 2 (♥)
Kartu setelah pertukaran 10: 2 3 10 J Q K
Jumlah pertukaran yang dilakukan : 10
PS D:\code\c tugas> █
```

2. Saat sedang gabut, Kobo membayangkan suatu skenario dimana hanya ada satu buah bidak, yaitu kuda (knight), yang berada di dalam papan catur berukuran 8×8 . Kobo penasaran dan ingin mengetahui posisi mana saja yang dapat dicapai oleh bidak kuda tersebut dalam sekali jalan apabila bidak tersebut berada pada posisi i, j dengan rincian $0 \leq i, j < 8$. Namun, karena Kobo tidak punya uang untuk membeli papan catur, ia ingin mensimulasikannya pada komputer dengan menggunakan bahasa C. Dalam simulasinya, ia ingin punya array 2D dengan nilai awal 0 di setiap index sebagai bidak caturnya. Kemudian, Kobo ingin memberi nilai 1 pada setiap posisi yang mungkin dilalui oleh bidak kuda tersebut dalam sekali jalan apabila bidak tersebut berada pada posisi i, j . Bantulah Kobo untuk membuat program tersebut dengan melanjutkan fungsi berikut: `void kobolmaginaryChess(int i, int j, int size, int *chessBoard){...}` Input terdiri dari satu baris yang berisi nilai i dan j yang dipisahkan tanda spasi. Output adalah output semua nilai di setiap baris dan kolom dari array berukuran 8×8 sesuai soal

Code

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

void kobolmaginaryChess(int i, int j, int size, int
*chessBoard) {
    for (int row = 0; row < size; row++) {
        for (int col = 0; col < size; col++) {
            *(chessBoard + row*size + col) = 0;
        }
    }

    int moves[8][2] = {{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2},
                        {1, -2}, {1, 2}, {2, -1}, {2, 1}};
    for (int k = 0; k < 8; k++) {
        int new_i = i + moves[k][0];
        int new_j = j + moves[k][1];
        if (new_i >= 0 && new_i < size && new_j >= 0 && new_j
< size) {
            *(chessBoard + new_i*size + new_j) = 1;
        }
    }
}

int main() {

    int i, j;
    scanf("%d %d", &i, &j);
```

	<pre> int size = 8; int *chessBoard = (int *)malloc(size * size * sizeof(int)); koboImaginaryChess(i, j, size, chessBoard); for (int row = 0; row < size; row++) { for (int col = 0; col < size; col++) { printf("%d ", *(chessBoard + row*size + col)); } printf("\n"); } free(chessBoard); return 0; } </pre>
Penjelasan	<ul style="list-style-type: none"> • Library yang di-import: <ul style="list-style-type: none"> - <stdio.h>: Digunakan untuk fungsi input-output standar seperti printf() dan scanf(). - <stdlib.h>: Digunakan untuk alokasi dan dealokasi memori dinamis menggunakan fungsi malloc() dan free(). • Definisi Fungsi koboImaginaryChess(): <ul style="list-style-type: none"> - Fungsi ini bertujuan untuk menginisialisasi papan catur dan menandai posisi-posisi yang dapat dicapai oleh bidak kuda dari posisi yang ditentukan (i, j). - Pertama, fungsi ini menginisialisasi papan catur dengan nilai 0 di setiap sel menggunakan nested loop. - Selanjutnya, fungsi ini mengecek semua kemungkinan gerakan kuda (sebanyak 8 kemungkinan) dan menandai posisi yang dapat dicapai oleh bidak kuda dengan nilai 1. • Fungsi main(): <ul style="list-style-type: none"> - Pengguna diminta untuk memasukkan posisi (i, j) dari bidak kuda. - Kemudian, dilakukan alokasi memori dinamis untuk papan catur menggunakan malloc(). - Fungsi koboImaginaryChess() dipanggil untuk menandai posisi-posisi yang dapat dicapai oleh bidak kuda pada papan catur. - Setelah itu, papan catur yang telah ditandai dicetak ke layar menggunakan nested loop. - Terakhir, memori yang dialokasikan untuk papan catur dibebaskan menggunakan free().

Output	<div><div>PROBLEMS OUTPUT DEBUG CONSOLE <u>TERMINAL</u></div><div>PS D:\code\c tugas> cd "d:\code\c tugas\" ; i 2 2 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 1 0 PS D:\code\c tugas> </div><div>PS D:\code\c tugas> cd "d:\code\c tugas\" 3 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 PS D:\code\c tugas> </div></div>