Tugas praktikum

Nama: Danendra Urdha Bhanu Cetta Harenndy

Nim : 1203230110

Tugas 1:

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

int convertToValue(char card) {
    if (card == 'J') return 11;
    else if (card == 'Q') return 12;
    else if (card == 'K') return 13;
    else if (card == 'A') return 1;
    else return card - '0';
}

int findMinIndex(int arr[], int start, int end) {
    int minIndex = start;
    for (int i = start + 1; i < end; ++i) {
        if (arr[i] < arr[minIndex]) {
            minIndex = i;
        }
    }
    return minIndex;
}

void swap(int arr[], int i, int j) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

int selectionSort(int arr[], int n) {
    int steps = 0;
    for (int i = 0; i < n - 1; ++i) {
        int minIndex = findMinIndex(arr, i, n);
        if (minIndex != i) {
            swap(arr, i, minIndex);
            steps++;
            printf("penukaran %d: ", steps);
            for (int j = 0; j < n; ++j) {
                printf("%d ", arr[j]);
```

```
            }
            printf("\n");
        }
    }
    return steps;
}

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

    int cards[n];
    printf("");
    for (int i = 0; i < n; ++i) {
        char card[3];
        scanf("%s", card);
        cards[i] = convertToValue(card[0]);
    }

    int steps = selectionSort(cards, n);

    printf("%d\n", steps);

    return 0;
}
```

OUTPUT:

```
PS C:\Users\danen> cd "c:\Users\danen\OneDrive\Desktop\latihan\" ; if ($?) { gcc fuckkalukataguamah.c -o fuckkalukataguamah } ; if ($?) { .\fuckkalukataguamah }
4
6 6 9 7
penukaran 1: 6 6 7 9
1
PS C:\Users\danen\OneDrive\Desktop\latihan> cd "c:\Users\danen\OneDrive\Desktop\latihan\" ; if ($?) { gcc fuckkalukataguamah.c -o fuckkalukataguamah } ; if ($?) {
 .\fuckkalukataguamah }
5
3 2 8 7 4
penukaran 1: 2 3 8 7 4
penukaran 2: 2 3 4 7 8
2
```

```
6
10 J Q K 3 2
penukaran 1: 2 J K 3 Q 10
penukaran 2: 2 3 J K Q 10
penukaran 3: 2 3 10 J K Q
penukaran 4: 2 3 10 J Q K
4
```

Penjelasan kode :

Pada Fungsi convertToValue berfungsi untuk mengubah karakter yang mewakili kartu (misalnya, 'J', 'Q', 'K') menjadi nilai numerik. Seperti contoh huruf Q berarti akan ter output angka 12 , J menjadi 11 , K menjadi 13

Pada fungsi ini saya menggunakan parameter card : yang berfungsi menjadi karakter yang akan mewakili kartu

Pada fungsi ini saya juga menggunakan pernyataan if untuk memeriksa karakter card seperti ini:

- Jika card sama dengan 'J', fungsi ini mengembalikan nilai 11.
- Jika card sama dengan 'Q', fungsi ini mengembalikan nilai 12.
- Jika card sama dengan 'K', fungsi ini mengembalikan nilai 13.
- Jika card sama dengan 'A', fungsi ini mengembalikan nilai 1.
- Jika card tidak termasuk dalam kasus-kasus di atas, fungsi ini mengasumsikan bahwa card adalah angka dan mengembalikan nilainya dikurangi '0'.

Pada Fungsi findMinIndex berfungsi mencari indeks elemen dengan nilai terkecil dalam rentang tertentu dari array.

Pada fungsi ini saya menggunakan parameter:

Arr :pointer ke array yang berisi elemen-elemen yang akan dicari

Start: indeks awal dari rentang pencarian

End : indeks akhir dari rentang pencarian

Pada Fungsi ini melakukan perulangan dari start + 1 hingga end - 1. Di setiap iterasi, fungsi ini membandingkan nilai elemen pada indeks saat ini (i) dengan nilai elemen pada minIndex.Jika nilai elemen pada i lebih kecil daripada nilai elemen pada minIndex, maka minIndex diperbarui dengan nilai i.Setelah perulangan selesai, fungsi ini mengembalikan nilai minIndex sebagai indeks elemen dengan nilai terkecil dalam rentang [start, end].

Pada Fungsi void swap berfungsi menukar dua elemen dalam array.

Di fungsi ini saya menggunakan parameter:

Arr: pointer ke array yang berisi elemen-elemen yang akan ditukar

I : indeks elemen pertama yang akan ditukar

J : indeks elemen kedua yang akan ditukar

Pada Fungsi ini menggunakan variabel temp untuk menyimpan nilai elemen pada arr[i].

Nilai elemen pada arr[j] disimpan di arr[i].

Nilai yang disimpan di temp (semula nilai arr[i]) disimpan di arr[j].

Fungsi swap membantu menukar dua elemen dalam array, yang berguna dalam algoritma pengurutan seperti selection sort.

Pada Fungsi int selectionSort berfungsi mengimplementasikan algoritma selection sort untuk mengurutkan array arr yang berisi n elemen.

Pada fungsi ini saya menggunakn parameter:

arr: pointer ke array yang berisi elemen-elemen yang akan diurutkan

n: jumlah elemen dalam array

pada Fungsi ini menginisialisasi variabel steps untuk menghitung jumlah pertukaran.

Fungsi ini melakukan perulangan dari elemen pertama hingga elemen kedua dari belakang. Di setiap iterasi, fungsi ini:

- Mencari indeks elemen dengan nilai terkecil di sisa array menggunakan fungsi findMinIndex.
- Jika elemen dengan nilai terkecil tidak berada pada posisi yang benar, menukar elemen tersebut dengan elemen di posisi saat ini menggunakan fungsi swap.
- Menambahkan 1 ke steps setiap kali terjadi pertukaran.
- Menampilkan urutan kartu setelah setiap pertukaran.

Setelah perulangan selesai, fungsi ini mengembalikan nilai steps sebagai jumlah minimal langkah pertukaran yang dilakukan.

Pada fungsi main saya memasukan hasill dari fungsi-fungsi yang telah saya program

Didalamnya terdapat tugas yang saya telah masukan diantaranya adalah

- Membaca Jumlah Kartu
- Membaca Nilai Kartu
- Mengurutkan Kartu
- Menampilkan Hasil

Tugas 2:

Kode:

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

void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
  int dx[] = {-2, -1, 1, 2, 2, 1, -1, -2};
  int dy[] = {1, 2, 2, 1, -1, -2, -2, -1};

  for (int k = 0; k < 8; k++) {
    int x = i + dx[k];
    int y = j + dy[k];

    if (x >= 0 && x < size && y >= 0 && y < size) {
      chessBoard[x * size + y] = 1;
    }
  }
}

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

  int size = 8;
  int *chessBoard = (int *)malloc(size * size * sizeof(int));
  if (chessBoard == NULL) {
    printf("Memory allocation failed.\n");
    return 1;
  }

  for (int row = 0; row < size; row++) {
    for (int col = 0; col < size; col++) {
      chessBoard[row * size + col] = 0;
    }
  }

  koboImaginaryChess(i, j, size, chessBoard);

  for (int row = 0; row < size; row++) {
    for (int col = 0; col < size; col++) {
      if (row == i && col == j) {
        printf("0 ");
      } else {
        printf("%d ", chessBoard[row * size + col]);
      }
    }
    printf("\n");
  }
```

```
  free(chessBoard);

  return 0;
}
```

Output:

```
PS C:\Users\danen> cd "c:\Users\danen\OneDrive\Desktop
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 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

```
PS C:\Users\danen> cd "c:\Users\danen\O
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
```

Penjelasan kode pada program:

Pada kode ini saya menambahkan berberapa fungsi,berikut adalah penjelasannya

-Fungsi koboImaginaryChess berfungsi menandai semua posisi pada papan catur yang dapat dicapai oleh kuda dari posisi awal yang diberikan.

Pada fungsi ini saya menggunakan parameter:

i: Baris awal kuda

j: Kolom awal kuda.

size: Ukuran papan catur (diasumsikan 8x8).

chessBoard: Pointer ke array integer yang mewakili papan catur.

Cara kerja fungsi ini:

Dua array dx dan dy menampung nilai offset untuk 8 kemungkinan gerakan kuda (2 langkah ke kanan/kiri, 2 langkah ke atas/bawah, dikombinasikan).

Pada Perulangan berfungsi untuk mengulangi semua 8 kemungkinan gerakan ksatria menggunakan array dx dan dy.

Di dalam perulangan, x dan y dihitung dengan menambahkan nilai dx[k] dan dy[k] ke posisi awal (i, j). Nilai ini mewakili kemungkinan posisi baru setelah kuda bergerak.

Pernyataan if memeriksa apakah posisi baru (x, y) berada dalam batas papan catur (0 <= x < size dan 0 <= y < size).

Jika posisi baru valid, elemen array chessBoard yang sesuai ditandai sebagai 1, menunjukkan bahwa kuda dapat mencapai posisi tersebut.

Elemen array chessBoard diakses dengan rumus chessBoard[x * size + y]. Berfungsi:

Mengalikan x dengan size untuk melompat ke baris yang benar.

Menambahkan y untuk mendapatkan offset kolom dalam baris tersebut.

-Pada fungsi main, saya memasukan hasil dari fungsi Fungsi koboImaginaryChess yang telah saya program

Berikut princianya:

- Membaca Posisi Awal Kuda:

Dua variabel i dan j dideklarasikan untuk menyimpan baris dan kolom awal kuda.Fungsi scanf digunakan untuk membaca nilai-nilai ini dari input pengguna.

- Menginisialisasi Papan Catur:

Sebuah variabel size diset ke 8, untuk papan catur 8x8.Pointer chessBoard dideklarasikan untuk menampung alamat array integer.Fungsi malloc digunakan untuk mengalokasikan memori secara dinamis untuk papan catur, dengan ukuran size * size * sizeof(int).Jika alokasi memori gagal, program akan menampilkan pesan error dan keluar.Perulangan for digunakan untuk menginisialisasi semua elemen array chessBoard menjadi 0.

- Menampilkan Papan Catur:

Perulangan for digunakan untuk mencetak papan catur.Di dalam perulangan:

-Jika posisi saat ini adalah posisi awal kuda, 0 dicetak.

-Jika tidak, nilai di array chessBoard pada posisi tersebut dicetak.

Karakter \n ditambahkan di akhir setiap baris untuk mencetak baris baru.

Sekian dan terima kasih