

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

Idil Haq Al Farisi

1203230070

Informatika-03-03

Program 1

```

... C praktikum1.c X
3 > C praktikum1.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void printCards(int *cards, int N, int swapNum) {
5      printf("Perurutan ke-%d: ", swapNum);
6      for(int i = 0; i < N; i++) {
7          if(cards[i] == 11) printf("J ");
8          else if(cards[i] == 12) printf("Q ");
9          else if(cards[i] == 13) printf("K ");
10         else printf("%d ", cards[i]);
11     }
12     printf("\n");
13 }
14
15 int main() {
16     int N, i, j, temp, swaps = 0;
17     char input[3];
18
19     scanf("%d", &N);
20     int *cards = malloc(N * sizeof(int));
21
22     for(i = 0; i < N; i++) {
23         scanf("%s", input);
24         if(input[0] == 'J') cards[i] = 11;
25         else if(input[0] == 'Q') cards[i] = 12;
26         else if(input[0] == 'K') cards[i] = 13;
27         else cards[i] = atoi(input);
28     }
29     for(i = 0; i < N-1; i++) {
30         int min_idx = i;
31         for(j = i+1; j < N; j++) {
32             if(cards[j] < cards[min_idx]) {
33                 min_idx = j;
34             }
35         }
36         if(min_idx != i) {
37             temp = cards[min_idx];
38             cards[min_idx] = cards[i];
39             cards[i] = temp;
40             swaps++;
41             printCards(cards, N, swaps);
42         }
43     }
44     printf(" %d\n", swaps);
45     free(cards);
46     return 0;
47 }
48

```

```
PS C:\Users\USER\Desktop\ALQ\3> cd "c:\Users\USER\Desktop\ALQ\3\" ; if ($?) { gcc praktikum1.c -o praktikum1 } ; if ($?) { .\praktikum1 }
4
6 6 9 7
Perurutan ke-1: 6 6 7 9
1
PS C:\Users\USER\Desktop\ALQ\3> cd "c:\Users\USER\Desktop\ALQ\3\" ; if ($?) { gcc praktikum1.c -o praktikum1 } ; if ($?) { .\praktikum1 }
5
3 2 8 7 4
Perurutan ke-1: 2 3 8 7 4
Perurutan ke-2: 2 3 4 7 8
2
PS C:\Users\USER\Desktop\ALQ\3> cd "c:\Users\USER\Desktop\ALQ\3\" ; if ($?) { gcc praktikum1.c -o praktikum1 } ; if ($?) { .\praktikum1 }
6
10 J K Q 3 2
Perurutan ke-1: 2 J K Q 3 10
Perurutan ke-2: 2 3 K Q J 10
Perurutan ke-3: 2 3 10 Q J K
Perurutan ke-4: 2 3 10 J Q K
4
PS C:\Users\USER\Desktop\ALQ\3> cd "c:\Users\USER\Desktop\ALQ\3\" ; if ($?) { gcc praktikum1.c -o praktikum1 } ; if ($?) { .\praktikum1 }
8
9 4 2 J K 8 4 Q
Perurutan ke-1: 2 4 9 J K 8 4 Q
Perurutan ke-2: 2 4 4 J K 8 9 Q
Perurutan ke-3: 2 4 4 8 K J 9 Q
Perurutan ke-4: 2 4 4 8 9 J K Q
Perurutan ke-5: 2 4 4 8 9 J Q K
5
PS C:\Users\USER\Desktop\ALQ\3>
```

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

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

Berfungsi untuk mengimpor library untuk fungsi fungsi standar dan alokasi memori di bahasa c

```
void printCards(int *cards, int N, int swapNum) {
    printf("Perurutan ke-%d: ", swapNum);
    for(int i = 0; i < N; i++) {
        if(cards[i] == 11) printf("J ");
        else if(cards[i] == 12) printf("Q ");
        else if(cards[i] == 13) printf("K ");
        else printf("%d ", cards[i]);
    }
    printf("\n");
}
```

Fungsi printCards Berfungsi untuk mencetak urutan dari kartu. Fungsi ini menerima tiga argumen: array cards yang berisi nilai kartu, N yang merupakan jumlah kartu, dan swapNum yang merupakan jumlah pertukaran yang telah dilakukan. Fungsi ini mencetak “J”, “Q”, atau “K” untuk kartu dengan nilai 11, 12, atau 13, dan mencetak nilai numerik untuk kartu lainnya.

```
int main() {
    int N, i, j, temp, swaps = 0;
    char input[3];
```

```
    scanf("%d", &N);
```

int main merupakan fungsi utama yang akan dijalankan terlebih dahulu Ketika program dijalankan. Baris ini juga mendeklarasikan variabel n , I, j temp, dan swaps. Scanf berfungsi untuk menerima inputan untuk jumlah kartu

```
int *cards = malloc(N * sizeof(int));
```

```
for(i = 0; i < N; i++) {
    scanf("%s", input);
```

```

        if(input[0] == 'J') cards[i] = 11;
        else if(input[0] == 'Q') cards[i] = 12;
        else if(input[0] == 'K') cards[i] = 13;
        else cards[i] = atoi(input);
    }

```

Array cards dialokasikan untuk menyimpan nilai kartu. Pengguna diminta untuk memasukkan nilai kartu. Jika pengguna memasukkan “J”, “Q”, atau “K”, nilai kartu diset ke 11, 12, atau 13. Jika pengguna memasukkan angka, nilai kartu diset ke angka tersebut.

```

for(i = 0; i < N-1; i++) {
    int min_idx = i;
    for(j = i+1; j < N; j++) {
        if(cards[j] < cards[min_idx]) {
            min_idx = j;
        }
    }
    if(min_idx != i) {
        temp = cards[min_idx];
        cards[min_idx] = cards[i];
        cards[i] = temp;
        swaps++;
        printCards(cards, N, swaps);
    }
}

```

Baris ini berfungsi untuk melakukan sortir seleksi(selection sort) pada array cards .Untuk setiap elemen dalam array, fungsi mencari elemen dengan nilai terkecil dalam subarray yang belum diurutkan dan menukarnya dengan elemen pertama dalam subarray tersebut. Jika pertukaran dilakukan, jumlah pertukaran (swaps) di increment (ditambah) dan kemudian dicetak.

```

printf(" %d\n", swaps);
free(cards);
return 0;
}

```

Baris ini berfungsi untuk mencetak jumlah pertukaran (sort) yang telah dijalankan program dan membebaskan memori yang dialokasikan untuk array cards kemudian program berhenti dengan status sukses (return 0)

Program 2

```
C praktikum1.c C praktikum2.c
3 > C praktikum2.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void printBoard(int size, int *chessBoard) {
5      for(int i = 0; i < size; i++) {
6          for(int j = 0; j < size; j++) {
7              printf("%d ", *((chessBoard+i*size) + j));
8          }
9          printf("\n");
10     }
11 }
12
13 void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
14     int moves[8][2] = {{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2}, {1, -2}, {1, 2}, {2, -1}, {2, 1}};
15     for(int k = 0; k < 8; k++) {
16         int new_i = i + moves[k][0];
17         int new_j = j + moves[k][1];
18         if(new_i >= 0 && new_i < size && new_j >= 0 && new_j < size) {
19             *((chessBoard+new_i*size) + new_j) = 1;
20         }
21     }
22 }
23
24 int main() {
25     int i, j;
26
27     scanf("%d %d", &i, &j);
28     int size = 8;
29     int *chessBoard = (int *)calloc(size*size, sizeof(int));
30     koboImaginaryChess(i, j, size, chessBoard);
31     printBoard(size, chessBoard);
32     free(chessBoard);
33     return 0;
34 }
35
```

```
1 #include <stdio.h>
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL
● PS C:\Users\USER\Desktop\ALQ\3> cd "c:\Users\USER\Desktop\ALQ\3\" ; if ($?) { gcc praktikum2.c -o praktikum2 } ; if ($?) { .\praktikum2 }
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\USER\Desktop\ALQ\3> cd "c:\Users\USER\Desktop\ALQ\3\" ; if ($?) { gcc praktikum2.c -o praktikum2 } ; if ($?) { .\praktikum2 }
3 3
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 C:\Users\USER\Desktop\ALQ\3>
```

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

Berfungsi untuk mengimpor library untuk fungsi fungsi standar dan alokasi memori di bahasa c

```

void printBoard(int size, int *chessBoard) {
    for(int i = 0; i < size; i++) {
        for(int j = 0; j < size; j++) {
            printf("%d ", *((chessBoard+i*size) + j));
        }
        printf("\n");
    }
}

```

Fungsi printBoard berfungsi untuk mencetak papan catur. Fungsi ini menerima 2 argumen, size dan chessboard. Size merepresentasikan ukuran papan catur dan chessboard yang merupakan array 2 dimensi yang merepresentasikan papan catur.

```

void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    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;
        }
    }
}

```

Fungsi koboImaginaryChess menandai semua posisi yang dapat dicapai oleh kuda catur dari posisi (i, j) pada papan catur. Fungsi ini menerima empat argumen: i dan j yang merupakan posisi awal kuda catur, size yang merupakan ukuran papan catur, dan chessBoard yang merupakan array dua dimensi yang merepresentasikan papan catur.

```

int main() {
    int i, j;

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

```

int main merupakan fungsi utama yang akan dijalankan terlebih dahulu Ketika program dijalankan. Baris ini juga mendeklarasikan variabel i dan j yang bertipe integer. Variabel I dan j digunakan untuk menyimpan posisi awal kuda catur. Scanf disini berfungsi untuk membaca inputan untuk dimasukkan ke variabel i dan j sebagai posisi awal kuda catur

```

    int size = 8;
    int *chessBoard = (int *)calloc(size*size, sizeof(int));
    koboImaginaryChess(i, j, size, chessBoard);
    printBoard(size, chessBoard);
    free(chessBoard);
    return 0;
}

```

Ukuran papan catur diset ke 8. Array chessBoard dialokasikan untuk menyimpan papan catur. Fungsi koboImaginaryChess dipanggil untuk menandai semua posisi yang dapat dicapai oleh kuda catur dari posisi (i, j). Kemudian, papan catur dicetak menggunakan fungsi printBoard. dan membebaskan memori yang dialokasikan untuk array chessBoard dan program berhenti dengan status sukses (return 0)