



POLITECHNIKA RZESZOWSKA

im. Ignacego Łukasiewicza

WYDZIAŁ MATEMATYKI I FIZYKI STOSOWANEJ

Karolina Wolska

Struktura danych typu kolejka FIFO

Projekt Algorytmny i Struktury Danych

kierunek studiów: Inżynieria i analiza danych

Rzeszów 2023

1. Treść zadania	3
2. Menu kolejki FIFO.....	4
3. Funkcja “dodaj”	5
3.1 Kod funkcji	5
3.2 Pseudokod funkcji “dodaj”	5
4. Funkcja “usuń”	6
4.1 Kod funkcji	6
4.2 Pseudokod funkcji “usuń”	6
5. Funkcja “wyświetl”	7
5.1 Kod funkcji	7
5.2 Pseudokod funkcji “wyswietl”	7
6. Funkcja “zliczanie”	8
6.1 Kod funkcji	8
6.2 Pseudokod funkcji “zliczanie”	8
7. Funkcja “szukanie”	9
7.1 Kod funkcji	9
7.2 Pseudokod funkcji “szukanie”	9

1. Treść zadania

Dokonaj implementacji struktury danych typu kolejka FIFO wraz z wszelkimi potrzebnymi operacjami charakterystycznymi dla tej struktury (inicjowanie struktury, dodawanie/usuwanie elementów, wyświetlanie elementów, zliczanie elementów/wyszukiwanie zadanego elementu itp.)

1) przyjąć, że podstawowym typem danych przechowywanym w elemencie struktury będzie struktura z jednym polem typu int

2) w funkcji main() przedstawić możliwości napisanej przez siebie biblioteki

3) kod powinien być opatrzony stosownymi komentarzami

2. Menu kolejki FIFO

Kolejka jest zdefiniowana jako liniowa struktura danych, która jest otwarta na obu końcach, a operacje są wykonywane w kolejności FIFO (First In First Out). Konsekwencją tej reguły jest to, że elementy są dodawane do kolejki na tylko jednym z jej końców, a usuwane wyłącznie na drugim.

W programie można wybrać 5 opcji działań na kolejce FIFO.

- 1. Dodawanie elementów z kolejki*
- 2. Usuwanie elementów z kolejki*
- 3. Wyświetlanie elementów kolejki*
- 4. Zliczanie elementów kolejki*
- 5. Wyszukiwanie zadanego elementu w kolejce*

Kolejka FIFO

```
1. Dodaj element
2. Usun element
3. Wypisz elementy
4. Zlicz elementy
5. Szukaj elementu
6. Wyjdz
Wybierz opcje: _
```

3. Funkcja “dodaj”

Wybierając funkcję “dodaj” program zapyta o liczbę, którą ma wprowadzić do kolejki zgodnie z zasadami kolejki FIFO. Jeśli elementów w kolejce jest już za dużo, program poinformuje o tym użytkownika.

3.1 Kod funkcji

```
void dodaj(){
    int element;
    if (koniec == LIMIT - 1){ // zabezpieczenie zeby nie przekroczyć limitu miejsc w kolejce
        printf("NIE UDALO SIE, za duzo elementow w kolejce\n\n");
    }
    else{
        if (poczatek == - 1) poczatek = 0;
        printf("Wpisz liczbe, ktora chcesz dodac do kolejki: ");
        scanf("%d", &element);
        printf("Dodano liczbe %d do kolejki\n\n", element);
        koniec++;
        queue[koniec] = element;
    }
}
```

```
1. Dodaj element
2. Usun element
3. Wypisz elementy
4. Zlicz elementy
5. Szukaj elementu
6. Wyjdz
```

Wybierz opcje:1

Wpisz liczbe, ktora chcesz dodac do kolejki: 55
Dodano liczbe 55 do kolejki

```
1. Dodaj element
2. Usun element
3. Wypisz elementy
4. Zlicz elementy
5. Szukaj elementu
6. Wyjdz
```

Wybierz opcje:1

NIE UDALO SIE, za duzo elementow w kolejce

3.2 Pseudokod funkcji “dodaj”

jeśli $k = \text{LIMIT} - 1$ to

 poinformuj o błędzie i zakończ działanie funkcji

jeśli $p = -1$ to

$p \leftarrow 0$

pobierz od użytkownika x

$k \leftarrow k + 1$

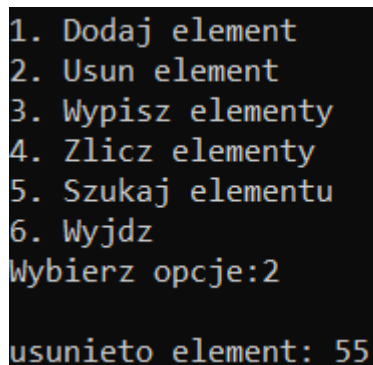
$\text{tab}[k] \leftarrow x$

4. Funkcja “usuń”

Wybierając funkcję “usuń” program usunie pierwszy element kolejki zgodnie z regułą “FIFO” i wyświetli usunięty element. Jeśli kolejka będzie pusta, program poinformuje o tym użytkownika.

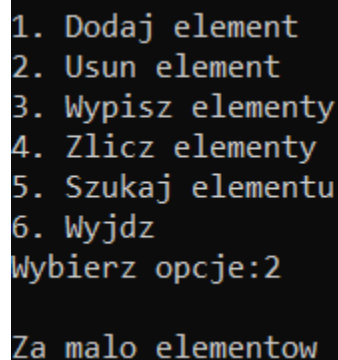
4.1 Kod funkcji

```
void usun(){
    if (poczatek == - 1 || poczatek > koniec){
        printf("Za malo elementow \n");
    }
    else{
        printf("usunieto element: %d\n", queue[poczatek]);
        koniec=koniec-1;
        for(i=0;i<=koniec;i++){
            queue[i]=queue[i+1];
        }
    }
}
```



```
1. Dodaj element
2. Usun element
3. Wypisz elementy
4. Zlicz elementy
5. Szukaj elementu
6. Wyjdz
Wybierz opcje:2

usunieto element: 55
```



```
1. Dodaj element
2. Usun element
3. Wypisz elementy
4. Zlicz elementy
5. Szukaj elementu
6. Wyjdz
Wybierz opcje:1

Za malo elementow
```

4.2 Pseudokod funkcji “usuń”

jeśli $p = -1$ lub $p > k$ to

wyświetl “pusta kolejka” i zakończ działanie funkcji

$k \leftarrow k-1$

$i \leftarrow 0$

dopóki $i \leq k$ wykonuj

$tab[i] \leftarrow tab[i+1]$

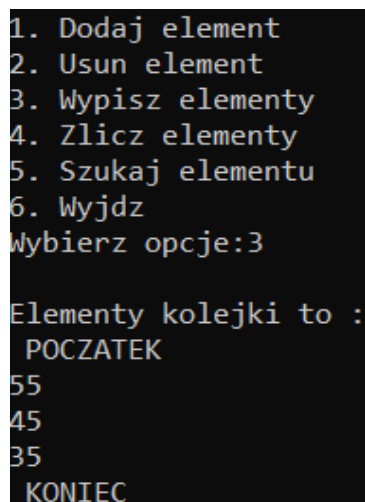
$i \leftarrow i+1$

5. Funkcja “wyświetl”

Wybierając funkcję wyświetl program wyświetli elementy znajdujące się w kolejce w odpowiedniej kolejności.

5.1 Kod funkcji

```
void wyswietl()  
{  
    int i;  
    if (poczatek == - 1)  
    {  
        printf("brak\n");  
    }  
    else  
    {  
        printf("Elementy kolejki to :\n POCZATEK\n");  
        for (i = poczatek; i <= koniec; i++)  
            printf("%d\n", queue[i]);  
    }  
    printf("KONIEC\n");  
}
```



```
1. Dodaj element  
2. Usun element  
3. Wypisz elementy  
4. Zlicz elementy  
5. Szukaj elementu  
6. Wyjdź  
Wybierz opcje:3  
  
Elementy kolejki to :  
POCZATEK  
55  
45  
35  
KONIEC
```

5.2 Pseudokod funkcji “wyswietl”

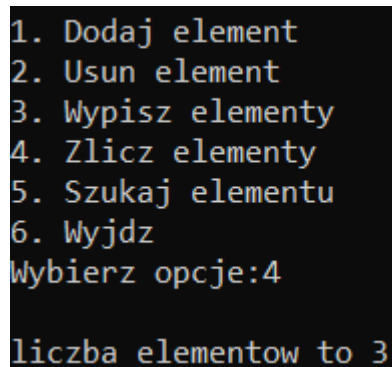
```
jeśli p = -1 to  
    wyświetl “brak” i zakończ działanie funkcji  
i <-- p  
dopóki i <= k wykonuj  
    wyświetl tab[i]  
    i <-- i + 1
```

6. Funkcja “zliczanie”

Wybierając tę funkcję program zliczy ile elementów znajduje się w tym momencie w kolejce i wyświetli tę liczbę.

6.1 Kod funkcji

```
void zliczanie()  
{  
    if(poczatek==-1) printf("0\n");  
    else{  
        int liczbael=koniec-poczatek+1;  
        printf("liczba elementow to %d\n", liczbael);  
    }  
}
```



```
1. Dodaj element  
2. Usun element  
3. Wypisz elementy  
4. Zlicz elementy  
5. Szukaj elementu  
6. Wyjdz  
Wybierz opcje:4  
  
liczba elementow to 3
```

6.2 Pseudokod funkcji “zliczanie”

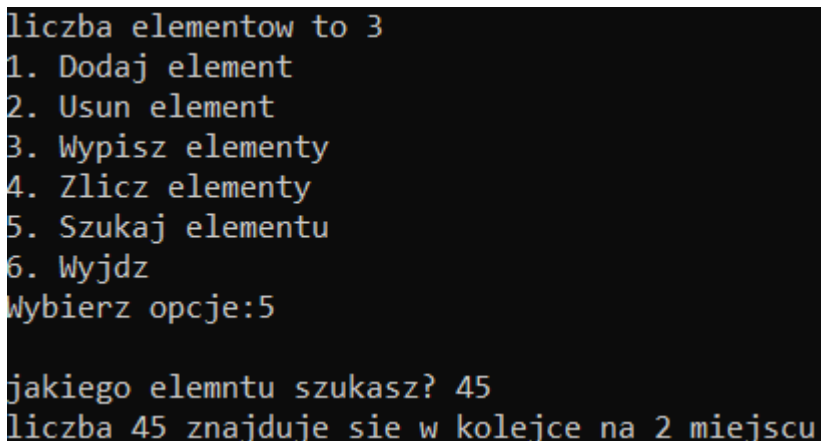
```
jeśli p = -1 to  
    wyświetl "0"  
l <-- k-p+1  
    wyświetl l
```


7. Funkcja “szukanie”

Wybierając tę funkcję program zapyta jakiej liczby w kolejce szukamy. Jeśli znajduje się ona w kolejce, program wypisze na którym miejscu występuje. Jeśli nie ma tej liczby w kolejce program poinformuje o tym użytkownika.

7.1 Kod funkcji

```
void szukanie()
{
    int p=0;
    printf("jakiego elemntu szukasz? ");
    scanf("%d", &k);
    for(i=poczatek; i<=koniec; i++){
        if(queue[i]==k){
            p++;
            printf("liczba %d znajduje sie w kolejce na %d miejscu\n", queue[i], i+1 );
        }
    }
    if(p==0){
        printf("brak tego elementu\n");
    }
}
```



```
liczba elementow to 3
1. Dodaj element
2. Usun element
3. Wypisz elementy
4. Zlicz elementy
5. Szukaj elementu
6. Wyjdz
Wybierz opcje:5

jakiego elemntu szukasz? 45
liczba 45 znajduje sie w kolejce na 2 miejscu
```

7.2 Pseudokod funkcji “szukanie”

Pobierz m

z <-- 0

i <-- p

dopóki i<=k wykonuj

jeśli tab[i] = m to

miejsce <-- i+1

z <-- z+1

jeśli z = 0 to

wyświetl “brak”