WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

**Wydział Cybernetyki**



Przedmiot:

**Algorytmy i struktury danych**

Laboratorium

**SPRAWOZDANIE**

|  |  |
| --- | --- |
| Autor: | Prowadzący: |
|  |  |
| Paweł Nowak | **Marcin Romaldowki** |

**Zadanie 1**.

Zaimplementuj algorytm a następnie napisz program, który na podstawie danych wejściowych utworzy listę (parametr m) i losowo wygeneruję liczby dla poszczególnych rekordów w tej liście a następnie przeniesie n-ty rekord (licząc od początku listy lub od jej końca) o k-pozycji. W implementowanym algorytmie użyj listy dwukierunkowej cyklicznej zaimplementowanej wskaźnikowo.



**Dane wejściowe (**podawane za pomocą klawiatury)**:**

m - długość listy;

n - n-ty rekord do przeniesienia

kier - kierunek poruszania (p - początek, k- koniec)

k - pozycja k

**Dane wyjściowe:**

Ekran: w pierwszej linii zostanie wyświetlona lista początkowa (każdy rekord\liczba oddzielona spacją) po wprowadzeniu dwóch pierwszych parametrów, dwie linie odstępu – lista po modyfikacji (każdy rekord\liczba oddzielona spacją).

Zaprezentuj algorytm w postaci:

* Listy kroków (sprawozdanie)
* Schematów blokowych (sprawozdanie)
* Kodu źródłowego w C++ (plik CPP, możliwość kompilacji w Dev-C++)

**Warunek zaliczenia zajęć:**

Terminowe przekazanie wyników prac:

* Sprawozdanie (jeśli wymagany)
* kod programu (jeśli wymagany)

jako spakowany (ZIP) załącznik do e-mail’a zatytułowanego :

Tytuł e-mail:

[ALS]-[NAZWA GRUPY] – [Nr zajęć Format: 00] - [Nazwisko i Imię]

ALS - I0X6S1 - 01 - Kowalczyk Mieczysław

**Lista kroków**

***Lista***

**Dane wejściowe:**

m - długość listy n - n-ty rekord do przeniesienia

kier - kierunek poruszania

k - pozycja k

**Dane wyjściowe:**

Ekran: w pierwszej linii zostanie wyświetlona lista początkowa (każdy rekord\liczba oddzielona spacją) po wprowadzeniu dwóch pierwszych parametrów, dwie linie odstępu – lista po modyfikacji (każdy rekord\liczba oddzielona spacją

1. **Początek programu**
2. **Zdefiniowanie 2 struktur : Listy i węzła (znajdują się w nich konstruktory domyślne)**
3. **Wywołanie instrukcji Insert :**

* Stworzenie pomocniczej zmiennej typu int value=0
* Lokowanie pamięci
* Wylosowanie liczby losowej
* Ustawienie wkaznika na kolejny element Listy
* Instukcja Insert będzie wykonywana dopóki value rekordów nie będzie wylosowanych i zapisanych na liście

1. **Wywołanie instrukcji Move**

* Dodanie 3 zmiennych typu int : Position, Direction, Shift
* Dodanie działania modulo po to aby przesunięcie nie wyszło poza zakres rekordów z listy
* Zadeklarowanie \*temp, \*temp2, \*temp3, \*par
* ***Jeżeli kier „p” :***
* Temp = head
* Zadeklarowanej zmiennej pomocniczej i;
* Pętla For
* Wybór pola ze strukury temp->next dalej to samo z par tyle że wskaźnik na poprzedni element
* Dochodzimy do momentu temp->next->prev= par
* Konstrukcja warunkowa if (temp== head)
* Head wskazuje na następny element
* Konstrukcja Warunkowa else if (temp==tail)
* Nasz „koniec” = par
* Ustawimy zmienne wskaźnikowe temp2= temp oraz temp3= head
* Konstrukcja warunkowa if (shift+position) == size
* Tail wskazuje na next = temp2; temp2 wskazuje na next = head; head wkazuje na prev=temp2
* Zwracamy (element wstawiony na krańcu listy)
* Pętla for (zmienna pomocnicza i typu int) --- temp 3 = temp 3 wskazuje na pole next
* Koniec pętli
* Par = temp 3 wskazuje na pole prev
* Temp 2 wskazuje na pole prev = par
* Temp 2 wskazuje na pole next = temp3
* Par wskazuje na pole next = temp2
* Wstawienie przed wybrany element
* Konstrukcja warunkowa if ( temp == head)
* To head = temp2
* ***ELSE ( tutaj jest kierunek odwrotny czyli od prawej do lewej)***
* Naszym Temp = tail
* Pętla for (dodanie pomocniczej zmiennej i typu int)
* Temp wskazuje na pole prev = temp
* Par = temp->prev
* Dochodzimy do momentu temp->next->prev= par
* Pętla if (temp==head)
* Head= temp wskazuje nanext
* Konstrukcja Warunkowa else if (temp==tail)
* Nasz „koniec” = par
* Ustawimy zmienne wskaźnikowe temp2= temp oraz temp3= tail
* Konstrukcja Warunkowa if((shift+ position )= size)
* Temp2 wskazuje na polenext = head
* Head = temp2
* Tail wksazuje na next = temp2
* Head wskazuje na prev = tail
* Zwracamy
* Pętla For (dodajemy zmienną pomocniczą i typu int)
* Temp3 = temp3 wskazuje na prev
* Koniec pętli For
* Konstrukcja wanukowa if (temp3==tail)
* Tail wskazuje na pole next = temp2
* Temp2 wskazujena pole prev=tail
* Co za tymidzie temp2=tail
* Tail wskazuje Na pole next = head
* Zwracamy
* Teraz : par = temp3 wskazuje na prev
* Temp 2 wskazuje na prev = par
* Temp2 wskazuje na next = temp3
* Par wskazuje na next = temp2

1. **Wywołanie instrukcji print**

* Node\*temp= head
* Pętla do while
* Wypisywanie od head, wskaźnik ustawiony na następny element listy
* Instrukcja będzie się wykonywać dopóki nie zostaną wypisane wszystkie elementy.

1. **Początek programu.**

**Schemat blokowy i kod źródłowy**

Schematy blokowe i kody źródłowe zostały załączone do niniejszego sprawozdania, nie zostały umieszczone w sprawozdaniu ze względu na ich wielkość.

**Wnioski**

Program bazuje na liście dwukierunkowej - cyklicznej. Zadaniem programu było wylosowanie n rekordów po czym wybranie jednego z nich a następnie wstawienie w dowolne miejsce na liście. W algorytmie trzeba było uwzględnić w którym kierunku chcieliśmy przesuwać element. Jedną z najważniejszych rzeczy było zachowanie cykliczności listy. Działanie modulo w tym przypadku bardzo ułatwiło to zadanie. W tej wersji programu lista może być nieskończenie długa – jednak długość ta może zostać ograniczona takimi rzeczami jak : Zasoby sprzętowe, ilość pamięci.