**ЗВІТ ДО ПР № 12-13**

*Виконав студент 35 групи*

*Рижій Ілля*

**Завдання 1.** Скласти програму Stack.cpp, реалізувати основні функції роботи зі стеком на прикладі масиву цілих чисел.

*Код програми:*

#include <stdio.h>

#include <stdlib.h>

#define N 10 // надання змінній N глобального значення

struct stack // оголошення структури

{

int elem[N];

int top;

};

typedef struct stack stack; // надання структірі псевдоніму для подальшого використання

void init(struct stack \*stk)

{

stk -> top = 0; // прирівнюємо елементи стеку до нуля

}

// додавання елементів у стек

void push(struct stack \*stk, int f)

{

if(stk->top < N) // якшщо розміри стеку менше вказаного розміру

{

stk->elem[stk->top] = f; // надаємо елементу введене значення

stk->top++; // переходимо до іншого елементу

}

else // якшщо розміри стеку бульше вказаного розміру

printf("count stack elements --> %d!\n", stk->top); // виведення на екран повідомлення про переповнення стеку

}

//видалення елементів стеку

int pop(struct stack \*stk)

{

int elem;

if((stk->top) > 0) // якщо розміри стеку більше нуля

{

stk->top--; // видаляємо останній елемент

elem = stk->elem[stk->top]; // надаємо змінній значення видаленого елементу

return elem; // повертаємо його значення

}

else // якщо розміри стеку дорівнює нулю

{

printf("Stack is NULL!\n"); // виведення на екран про порожність стеку

return 0;

}

}

// визначення пустоти стеку

int isempty (struct stack \*stk)

{

if((stk->top) == 0) //якщо стек порожній

return 1; // повертає занчення true

else

return 0; // повертає значення false

}

//виведення елементів стеку

void stkPrint(struct stack \*stk)

{

int i;

i = stk->top;

if(isempty(stk) == 1) // якщо стек порожній

return; //нічого не відбувається

do

{

i--;

printf("%d->", stk->elem[i]); //виведення елементів стеку на екран

}

while(i > 0);

}

//опис меню

void instructions(void)

{

printf("\n 1. Add in top \

\n 2. Delete \

\n 3. Show stack \

\n 0. Exit \n");

}

int main()

{

stack stack; //ініціалізація стеку в головній функції

init(&stack);

while(1)

{

instructions();

int c, p;

scanf("%d",&c);

system("cls");

switch(c){

case 1:

printf("\nInput data->");

scanf("%d", &p);

push(&stack, p); // виклик функції push, для додавання елементів стеку

break;

case 2:

printf("\n%d was deleted\n", pop(&stack)); // виведення на екран інформація про те, який елемент було видалено

break;

case 3:

stkPrint(&stack); // виклик функції stkPrint для виведення елементів стеку

break;

case 0:

exit(0);//вихід з программи

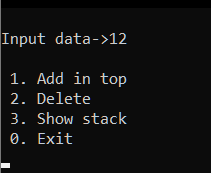
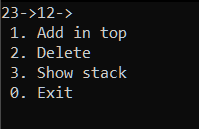
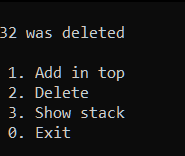
break;

}

}

return 0;

}

*Скріншоти роботи функцій введення, видалення та виведення елементів стеку*

**Завдання 2.** Скласти програму Queue.cpp, реалізувати основні функції роботи з чергою на прикладі масиву дійсних чисел.

Код програми:

#include <stdio.h>

#include <stdlib.h>

#define N 10

// оголошення структури

struct queue

{

int qu[N];

int rear, frnt;

};

typedef struct queue queue;

void init(struct queue \*q)

{

q->frnt = 1;

q->rear = 0; // прирінюємо елементи черги до нуля

}

// додавання елементів у черги

void insert(struct queue \*q, int x)

{

if(q->rear < N-1) // якщо розміри черги менше вказаного розміру

{

q->rear++; // переходимо до іншого елементу

q->qu[q->rear] = x; // надаємо елементу введене значення

}

else // якщо розмір черги більше вказаного розміру

printf("count queue elements -> %d!\n", q->rear); // виведення на екран повідомлення про переповнення черги

}

//видалення елементів черги

int pop(struct queue \*q)

{

int x;

if(isempty(q) == 1)

{

printf("Queue is NULL!");

return 0;

}

x = q->qu[q->frnt];

q->frnt++;

return x;

}

// визначення пустоти черги

int isempty(struct queue \*q)

{

if((q->rear)==0) //якщо черга порожня

return 1;

else

return 0;

}

//виведення елементів черги

void stkPrint(struct queue \*q)

{

int i;

i = q->frnt;

if(isempty(q) == 1) // якщо черга порожня

return; //нічого не відбувається

for(i = q->frnt; i <= q->rear; i++)

{

printf("%d<-", q->qu[i]); // виведення елементів черги на екран

}

}

// опис меню

void instructions(void)

{

printf ("\n 1. Add \

\n 2. Delete \

\n 3. Show queue \

\n 0. Exit \n");

}

int main()

{

queue queue; // ініціалізація черги в головній функції

init(&queue); // виклик функції init

while(1)

{

instructions(); //виведення

int c, p;// ініціалізація додаткових змінних

scanf ("%d",&c);

system("cls");

switch (c){

case 1:

printf("\nInput data->");

scanf("%d", &p); // введення числа для черги

insert(&queue,p); // виклик функції insert, для додавання елементів черги

break;

case 2:

printf("\n%d was deleted",pop(&queue)); // виведення на екран інформація про те, який елемент було видалено

break;

case 3:

stkPrint(&queue); // виклик функції stkPrint для виведення елементів черги

break;

case 0:

exit (0);//вихід з программи

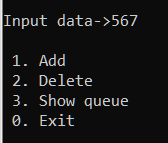
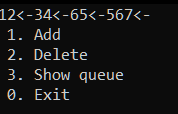
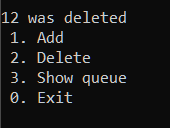
break;

}

}

return 0;

}

*Скріншоти роботи функцій введення, видалення та виведення елементів черги*

**Висновки:**  під час практичної роботи узагальнили знання про стеки та черги.