

Отчет по лабораторной работе № 25-26 по курсу «Практикум на ЭВМ»

Студент группы М8О-105Б-21 Минеева Светлана Алексеевна, № по списку 14

Контакты e-mail: svetlana.mineewa2003@yandex.ru

Работа выполнена: «23» мая 2022 г.

Преподаватель: Вячеслав Константинович Титов каф. 805

Отчет сдан «23» мая 2022 г., итоговая оценка _____

Подпись преподавателя _____

- Тема:** Лабораторная работа №25: Автоматизация сборки программ модульной структуры на языке Си с использованием утилиты make. Лабораторная работа №26: Абстрактные типы данных. Рекурсия. Модульное программирование на языке Си.
- Цель работы:** Лабораторная работа №25: Изучить принцип работы утилиты make. Лабораторная работа №26: Составить и отладить модуль определений и модуль реализации по заданной схеме модуля определений для абстрактного (пользовательского) типа данных (стека, очереди, списка или дека, в зависимости от варианта задания). Составить программный модуль, сортирующий экземпляр указанного абстрактного типа данных заданным методом, используя только операции, импортированные из модуля UUDT.
- Задание (вариант № 14):** Лабораторная работа №25: Автоматизировать сборку программ модульной структуры на языке Си с использованием утилиты make.
Лабораторная работа №26:
Тип данных: 1. Стек.
Процедура: 4. Поиск в очереди, списке, стеке или деке двух элементов, идущих подряд, первый из которых больше второго. Если такие элементы найдены, их перестановка.
Метод: 4. Сортировка методом пузырька.
- Оборудование (лабораторное):**
ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор 2,9 GHz 2-ядерный процессор Intel Core i5 с ОП 8 Гб, НМД 500 Гб. Монитор 13,3-дюймовый (2560 x 1600).

Другие устройства _____

5. Программное обеспечение (лабораторное):

Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____

Прикладные системы и программы _____

Местонахождение и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства UNIX, наименование Terminal версия 2.10

интерпретатор команд bash версия 3.2.

Система программирования _____ версия _____

Редактор текстов emacs версия 27.2-2

Утилиты операционной системы cat, ls, make, gcc и другие

Прикладные системы и программы _____

Местонахождение и имена файлов программ и данных на домашнем компьютере _____

6. Идея, метод, алгоритм решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Лабораторная работа №25:

Алгоритм работы:

Нам нужно скомпилировать программу из трех модулей при помощи makefile, для этого подробнее рассмотрим наши модули. Main.cpp и lab26.cpp – это основные модули, которые должны образовать скомпилированную программу, но они обе зависят от lab26.h, значит при преобразовании их в файлы с расширением .o нужно подключать к ним этот файл.

Лабораторная работа №26:

Описание сортировки:

Создаем вспомогательный стек, в который будем по очереди добавлять элементы из основного, если добавляемый элемент меньше или равен верхнему элементу вспомогательного стека. Если добавляемый элемент больше верхнего, то вызывается процедура Search_Swap, которая находит нужное место в стеке для этого элемента. Процедура принимает два стека из процедуры сортировки – исходный и вспомогательный, но уже преобразованные. Запоминаем первый элемент m исходного стека и удаляем его из стека. Начинаем идти по вспомогательному стеку, сравнивая эти элементы с m, перемещая элементы из вспомогательного в исходный стек, пока m не будет больше или равен верхнему элементу вспомогательного стека. Если мы достигли этого условия, то помещаем m в вспомогательный стек.

7. Сценарий выполнения работы [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

Лабораторная работа №25:

```
program: main.o lab26.o
        g++ -o program main.o lab26.o
main.o: main.cpp lab26.h
        g++ -c main.cpp
lab26.o: lab26.cpp lab26.h
        g++ -c lab26.cpp
```

Так как стек заполняется рандомно, тесты просчитаны уже после выполнения пункта «распечатка протокола».

Тесты (проверяем, что созданный файл program при запуске работает корректно):

1) [7 9 3 8 0 2 4 8]

Отсортированный: [0 2 3 4 7 8 8 9]

Лабораторная работа №26:

Реализация отображением на массив:

1) main.cpp

```
#include "lab26.h"
int main(){
    Stack S, S1, S2; Init(S); Tvalue v; int p=1,n,k;
    printf("\nChoose action:"
        "\n1. Create Random Stack\n2. Print Stack\n3. Size of Stack"
        "\n4. Insert of Stack"
        "\n5. Delete from front of Stack"
        "\n6. Sorting Stack\n7. Clear Stack"
        "\n0. Exit");
    while(p){
        printf("\nInput number => "); scanf("%d",&p);
        if(p==1)
        { printf("Input number of elements => "); scanf("%d",&n);
          Init(S1);
```

```

for(int i=0;i<n/2;i++){v=rand()%10; Push(S1,v);}
Display(S1);
Init(S2);
for(int i=0;i<n-n/2;i++){v=rand()%10; Push(S2,v);}
Display(S2); Cat(S1,S2); Append(S,S1); Display(S);}
if(p==2) Display(S);
if(p==3) { n=Size(S); printf("\nSize of Stack = %d\n",n); }
if(p==4) { printf("\nInput value of element => "); scanf("%d",&v); Push(S,v);}
if(p==5) { v=Pop(S); printf("\nElement = %d is deleted\n",v); }
if(p==6) Sort(S);
if(p==7) Init(S);
if(p==0) break;}}

```

2) lab26.h

```

#ifndef LAB26_H
#define LAB26_H
#include<stdio.h>
#include<stdlib.h>
#define N 100
#define Tvalue int

struct Stack{int first; Tvalue body[N];};

void Init(Stack &S);
int Empty(Stack S);
void Push(Stack &S, Tvalue V);
Tvalue Pop(Stack &S);
Tvalue Top(Stack S);
int Size(Stack S);
void Display(Stack S);
void Cat(Stack &S1, Stack &S2);
void Append(Stack &S1, Stack &S2);
void Search_Swap(Stack &S, Stack &S1);
void Sort(Stack &S);

#endif

```

3) lab26.cpp

```

#include "lab26.h"

void Init(Stack &S){S.first=0; }

int Empty(Stack S){return S.first==0;}

void Push(Stack &S, Tvalue V){
    if(S.first==N) printf("STACK IS OVERFLOW");
    else S.body[S.first++]=V;}

Tvalue Pop(Stack &S){
    if(Empty(S)) printf("STACK IS EMPTY");
    else return S.body[--S.first];}

Tvalue Top(Stack S){
    if(Empty(S)) printf("STACK IS EMPTY");
    else return S.body[S.first-1];}

int Size(Stack S){return S.first;}

void Display(Stack S){
    printf("\n[ ");
    for(int i=0;i<S.first;i++) printf("%d ",S.body[i]);
    printf("]\n");}

```

```

void Cat(Stack &S1, Stack &S2){
    Tvalue V;
    if(!Empty(S2)){ V=Pop(S2); Cat(S1,S2); Push(S1,V);}}

void Append(Stack &S1, Stack &S2){
    Tvalue V;
    if(Empty(S1)) Cat(S1,S2);
    else { V=Pop(S1); Append(S1,S2); Push(S1,V);}}

void Search_Swap(Stack &S, Stack &S1){
    bool swap=0; int f=Pop(S), s=Top(S1);
    while(f>s && !Empty(S1)){ s=Pop(S1); Push(S,s);
    if (!Empty(S1)) s=Top(S1);
    swap=1;}
    if(swap){ Push(S1,f); swap=0;}}

void Sort(Stack &S){
    Stack S1; Tvalue V,V1;
    Init(S1); int k=1;
    while (!Empty(S)){
        int e = Pop(S);
        if(Empty(S1) || Top(S1)>=e) Push(S1,e);
        else{ Push(S,e); Search_Swap(S,S1);}}
    while (!Empty(S1)) Push(S,Pop(S1));
    Init(S1);}

```

Реализация в виде динамической структуры:

1) *main1.cpp*

```

#include "lab261.h"

int main(){
    Stack S, S1, S2; Init(S); Tvalue v; int p=1,n,k;
    printf("\nChoose action:"
        "\n1. Create Random Stack\n2. Print Stack\n3. Size of Stack"
        "\n4. Insert of Stack"
        "\n5. Delete from front of Stack"
        "\n6. Sorting Stack\n7. Clear Stack"
        "\n0. Exit");
    while(p){
        printf("\nInput number => "); scanf("%d",&p);
        if(p==1)
        { printf("Input number of elements => "); scanf("%d",&n);
        Init(S1);
        for(int i=0;i<n/2;i++){v=rand()%10; Push(S1,v);}
        Display(S1);
        Init(S2);
        for(int i=0;i<n-n/2;i++){v=rand()%10; Push(S2,v);}
        Display(S2); Cat(S1,S2); Append(S,S1); Display(S);}
        if(p==2) Display(S);
        if(p==3) { n=Size(S); printf("\nSize of Stack = %d\n",n); }
        if(p==4) { printf("\nInput value of element => "); scanf("%d",&v); Push(S,v);}
        if(p==5) { v=Pop(S); printf("\nElement = %d is deleted\n",v); }
        if(p==6) Sort(S);
        if(p==7) Init(S);
        if(p==0) break;}}

```

2) *lab261.h*

```

#ifndef LAB261_H
#define LAB261_H
#include<stdio.h>
#include<stdlib.h>
#define N 100

```

```

#define Tvalue int

struct El_St{ Tvalue V; El_St *next;};
struct Stack{ El_St *first;int size;};

void Init(Stack &S);
int Empty(Stack S);
void Push(Stack &S, Tvalue V);
Tvalue Pop(Stack &S);
Tvalue Top(Stack S);
int Size(Stack S);
void Cat(Stack &S1, Stack &S2);
void Append(Stack &S1, Stack &S2);
void Display(Stack S);
void Search_Swap(Stack &S, Stack &S1);
void Sort(Stack &S);

#endif

```

3) lab261.cpp

```

#include "lab261.h"

void Init(Stack &S){ S.first=0; S.size=0;}

int Empty(Stack S){return S.first==0;}

void Push(Stack &S, Tvalue V){
    El_St *t=new El_St;
    t->next=S.first; S.first=t;
    S.first->V=V; S.size++;}

Tvalue Pop(Stack &S){
    if(Empty(S)) printf("STACK IS EMPTY");
    else { Tvalue V=S.first->V; S.first=S.first->next; S.size--; return V;}}

Tvalue Top(Stack S){
    if(Empty(S)) printf("STACK IS EMPTY");
    else return S.first->V;}

int Size(Stack S){return S.size;}

void Cat(Stack &S1, Stack &S2){
    Tvalue V;
    if(!Empty(S2)){ V=Pop(S2); Cat(S1,S2); Push(S1,V);}}

void Append(Stack &S1, Stack &S2){
    Tvalue V;
    if(Empty(S1)) Cat(S1,S2);
    else{ V=Pop(S1); Append(S1,S2); Push(S1,V);}}

void Display(Stack S){
    if(Empty(S)) printf("\nSTACK IS EMPTY\n");
    else{
        printf("\n["); El_St *t=S.first;
        while(t) { printf("%d ",t->V); t=t->next;}
        printf("]\n");}}

void Search_Swap(Stack &S, Stack &S1){
    bool swap=0; int f=Pop(S), s=Top(S1);
    while(f>s && !Empty(S1)){ s=Pop(S1); Push(S,s);
        if (!Empty(S1)) s=Top(S1);
        swap=1;}
    if(swap){ Push(S1,f); swap=0;}}

void Sort(Stack &S){

```

```

Stack S1; Tvalue V,V1;
Init(S1); int k=1;
while (!Empty(S)){
    int e = Pop(S);
    if(Empty(S1) || Top(S1)>=e) Push(S1,e);
    else{ Push(S,e); Search_Swap(S,S1);}
    while (!Empty(S1)) Push(S,Pop(S1));
Init(S1);}

```

Так как стек заполняется рандомно, тесты просчитаны уже после выполнения пункта «распечатка протокола».

Тесты:

- 1) [7 9 3 8 0 2]
Отсортированный: [0 2 3 7 8 9]
- 2) [4 8 3 9 0 5 2 2 7 3]
Отсортированный: [0 2 2 3 3 4 5 7 8 9]
- 3) [8 4 2 0 8 3 9 7]
Отсортированный: [0 2 3 4 7 8 8 9]
- 4) [9 9 3 2 0 9 7 3 7 2 2 5 0 9 3]
Отсортированный: [0 0 2 2 2 3 3 3 5 7 7 9 9 9 9]

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

Лабораторная работа №25:

Last login: Mon May 23 22:02:18 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit <https://support.apple.com/kb/HT208050>.
MacBook-Pro-MacBook:~ macbookpro\$ cat zag.txt

```
*****
*   Минеева Светлана Алексеевна   *
*           M80-105B-21             *
*   Лабораторная работа №25       *
*****
```

MacBook-Pro-MacBook:~ macbookpro\$ ls |tail -8

```
lab26.cpp
lab26.h
lab261.cpp
lab261.h
main.cpp
main1.cpp
pslog_20200707_123036.log
zag.txt
```

MacBook-Pro-MacBook:~ macbookpro\$ cat main.cpp

```
#include "lab26.h"
int main(){
    Stack S, S1, S2; Init(S); Tvalue v; int p=1,n,k;
    printf("\nChoose action:"
        "\n1. Create Random Stack\n2. Print Stack\n3. Size of Stack"
        "\n4. Insert of Stack"
        "\n5. Delete from front of Stack"
        "\n6. Sorting Stack\n7. Clear Stack"
        "\n0. Exit");
    while(p){
        printf("\nInput number => "); scanf("%d",&p);
        if(p==1)
        { printf("Input number of elements => "); scanf("%d",&n);
          Init(S1);
          for(int i=0;i<n/2;i++){v=rand()%10; Push(S1,v);}
          Display(S1);
          Init(S2);
          for(int i=0;i<n-n/2;i++){v=rand()%10; Push(S2,v);}
          Display(S2); Cat(S1,S2); Append(S,S1); Display(S);}
        if(p==2) Display(S);
        if(p==3) { n=Size(S); printf("\nSize of Stack = %d\n",n); }
        if(p==4) { printf("\nInput value of element => "); scanf("%d",&v); Push(S,v);}
        if(p==5) { v=Pop(S); printf("\nElement = %d is deleted\n",v); }
        if(p==6) Sort(S);
        if(p==7) Init(S);
        if(p==0) break;}}
MacBook-Pro-MacBook:~ macbookpro$ cat lab26.h
```

MacBook-Pro-MacBook:~ macbookpro\$ cat lab26.h

```
#ifndef LAB26_H
#define LAB26_H
#include<stdio.h>
#include<stdlib.h>
#define N 100
#define Tvalue int

struct Stack{int first; Tvalue body[N];};

void Init(Stack &S);
int Empty(Stack S);
void Push(Stack &S, Tvalue V);
Tvalue Pop(Stack &S);
Tvalue Top(Stack S);
int Size(Stack S);
void Display(Stack S);
void Cat(Stack &S1, Stack &S2);
void Append(Stack &S1, Stack &S2);
void Search_Swap(Stack &S, Stack &S1);
void Sort(Stack &S);

#endif
```

```

MacBook-Pro-MacBook:~ macbookpro$ cat lab26.cpp
#include "lab26.h"

void Init(Stack &S){S.first=0; }

int Empty(Stack S){return S.first==0;}

void Push(Stack &S, Tvalue V){
    if(S.first==N) printf("STACK IS OVERFLOW");
    else S.body[S.first++]=V;}

Tvalue Pop(Stack &S){
    if(Empty(S)) printf("STACK IS EMPTY");
    else return S.body[--S.first];}

Tvalue Top(Stack S){
    if(Empty(S)) printf("STACK IS EMPTY");
    else return S.body[S.first-1];}

int Size(Stack S){return S.first;}

void Display(Stack S){
    printf("\n[ ");
    for(int i=0;i<S.first;i++) printf("%d ",S.body[i]);
    printf("]\n");}

void Cat(Stack &S1, Stack &S2){
    Tvalue V;
    if(!Empty(S2)){ V=Pop(S2); Cat(S1,S2); Push(S1,V);}}

void Append(Stack &S1, Stack &S2){
    Tvalue V;
    if(Empty(S1)) Cat(S1,S2);
    else { V=Pop(S1); Append(S1,S2); Push(S1,V);}}

void Search_Swap(Stack &S, Stack &S1){
    bool swap=0; int f=Pop(S), s=Top(S1);
    while(f>s && !Empty(S1)){ s=Pop(S1); Push(S,s);
        if (!Empty(S1)) s=Top(S1);
        swap=1;}
    if(swap){ Push(S1,f); swap=0;}}

void Sort(Stack &S){
    Stack S1; Tvalue V,V1;
    Init(S1); int k=1;
    while (!Empty(S)){
        int e = Pop(S);
        if(Empty(S1) || Top(S1)>=e) Push(S1,e);
        else{ Push(S,e); Search_Swap(S,S1);}}
    while (!Empty(S1)) Push(S,Pop(S1));
    Init(S1);}

```

```

MacBook-Pro-MacBook:~ macbookpro$ cat >>Makefile
program: main.o lab26.o
    g++ -o program main.o lab26.o
main.o: main.cpp lab26.h
    g++ -c main.cpp
lab26.o: lab26.cpp lab26.h
    g++ -c lab26.cpp
MacBook-Pro-MacBook:~ macbookpro$ make -f Makefile
g++ -c main.cpp
g++ -c lab26.cpp
g++ -o program main.o lab26.o
MacBook-Pro-MacBook:~ macbookpro$ ls
Creative Cloud Files      kd.py.save
Desktop                   lab26.cpp
Documents                 lab26.h
Downloads                 lab26.o
IACPLHttpTransferClient  lab261.cpp
Library                   lab261.h
Makefile                  main.cpp
Movies                    main.o
Music                     main1.cpp
Pictures                  program
Public                    pslog_20200707_123036.log
PycharmProjects           zag.txt

```


MacBook-Pro-MacBook:~ macbookpro\$./program

Choose action:

1. Create Random Stack
2. Print Stack
3. Size of Stack
4. Insert of Stack
5. Delete from front of Stack
6. Sorting Stack
7. Clear Stack
0. Exit

Input number => 1

Input number of elements => 8

[7 9 3 8]

[0 2 4 8]

[7 9 3 8 0 2 4 8]

Input number => 6

Input number => 2

[0 2 3 4 7 8 8 9]

Input number => 0

MacBook-Pro-MacBook:~ macbookpro\$

Лабораторная работа №26:

Last login: Mon May 23 21:05:04 on ttys000

The default interactive shell is now zsh.

To update your account to use zsh, please run `chsh -s /bin/zsh`.

For more details, please visit <https://support.apple.com/kb/HT208050>.

MacBook-Pro-MacBook:~ macbookpro\$ cat zag.txt

```
*****
* Минеева Светлана Алексеевна *
*      М80-105Б-21      *
* Лабораторная работа №26 *
*****
```

MacBook-Pro-MacBook:~ macbookpro\$ ls |tail -8

```
lab26.cpp
lab26.h
lab261.cpp
lab261.h
main.cpp
main1.cpp
pslog_20200707_123036.log
zag.txt
```

MacBook-Pro-MacBook:~ macbookpro\$ cat main.cpp

```
#include "lab26.h"
int main(){
    Stack S, S1, S2; Init(S); Tvalue v; int p=1,n,k;
    printf("\nChoose action:"
        "\n1. Create Random Stack\n2. Print Stack\n3. Size of Stack"
        "\n4. Insert of Stack"
        "\n5. Delete from front of Stack"
        "\n6. Sorting Stack\n7. Clear Stack"
        "\n0. Exit");
    while(p){
        printf("\nInput number => "); scanf("%d",&p);
        if(p==1)
        { printf("Input number of elements => "); scanf("%d",&n);
          Init(S1);
          for(int i=0;i<n/2;i++){v=rand()%10; Push(S1,v);}
          Display(S1);
          Init(S2);
          for(int i=0;i<n-n/2;i++){v=rand()%10; Push(S2,v);}
          Display(S2); Cat(S1,S2); Append(S,S1); Display(S);}
        if(p==2) Display(S);
        if(p==3) { n=Size(S); printf("\nSize of Stack = %d\n",n); }
        if(p==4) { printf("\nInput value of element => "); scanf("%d",&v); Push(S,v);}
```

```

    if(p==5) { v=Pop(S); printf("\nElement = %d is deleted\n",v); }
    if(p==6) Sort(S);
    if(p==7) Init(S);
    if(p==0) break;}}
MacBook-Pro-MacBook:~ macbookpro$ cat lab26.h
#ifndef LAB26_H
#define LAB26_H
#include<stdio.h>
#include<stdlib.h>
#define N 100
#define Tvalue int

struct Stack{int first; Tvalue body[N];};

void Init(Stack &S);
int Empty(Stack S);
void Push(Stack &S, Tvalue V);
Tvalue Pop(Stack &S);
Tvalue Top(Stack S);
int Size(Stack S);
void Display(Stack S);
void Cat(Stack &S1, Stack &S2);
void Append(Stack &S1, Stack &S2);
void Search_Swap(Stack &S, Stack &S1);
void Sort(Stack &S);

#endif
MacBook-Pro-MacBook:~ macbookpro$ cat lab26.cpp
#include "lab26.h"

void Init(Stack &S){S.first=0; }

int Empty(Stack S){return S.first==0;}

void Push(Stack &S, Tvalue V){
    if(S.first==N) printf("STACK IS OVERFLOW");
    else S.body[S.first++]=V;}

Tvalue Pop(Stack &S){
    if(Empty(S)) printf("STACK IS EMPTY");
    else return S.body[--S.first];}

Tvalue Top(Stack S){
    if(Empty(S)) printf("STACK IS EMPTY");
    else return S.body[S.first-1];}

int Size(Stack S){return S.first;}

void Display(Stack S){
    printf("\n[ ");
    for(int i=0;i<S.first;i++) printf("%d ",S.body[i]);
    printf("]\n");}

void Cat(Stack &S1, Stack &S2){
    Tvalue V;
    if(!Empty(S2)){ V=Pop(S2); Cat(S1,S2); Push(S1,V);}}

void Append(Stack &S1, Stack &S2){
    Tvalue V;
    if(Empty(S1)) Cat(S1,S2);
    else { V=Pop(S1); Append(S1,S2); Push(S1,V);}}

void Search_Swap(Stack &S, Stack &S1){
    bool swap=0; int f=Pop(S), s=Top(S1);
    while(f>s && !Empty(S1)){ s=Pop(S1); Push(S,s);
        if (!Empty(S1)) s=Top(S1);
        swap=1;}
    if(swap){ Push(S1,f); swap=0;}}

void Sort(Stack &S){
    Stack S1; Tvalue V,V1;
    Init(S1); int k=1;
    while (!Empty(S)){
        int e = Pop(S);
        if(Empty(S1) || Top(S1)>=e) Push(S1,e);

```

```

        else{ Push(S,e); Search_Swap(S,S1);}}
        while (!Empty(S1)) Push(S,Pop(S1));
    Init(S1);}
MacBook-Pro-MacBook:~ macbookpro$ g++ -o lab26 lab26.cpp main.cpp
MacBook-Pro-MacBook:~ macbookpro$ ls |tail -9
lab26
lab26.cpp
lab26.h
lab261.cpp
lab261.h
main.cpp
main1.cpp
pslog_20200707_123036.log
zag.txt
MacBook-Pro-MacBook:~ macbookpro$ ./lab26

```

Choose action:

1. Create Random Stack
2. Print Stack
3. Size of Stack
4. Insert of Stack
5. Delete from front of Stack
6. Sorting Stack
7. Clear Stack
0. Exit

Input number => 1

Input number of elements => 6

[7 9 3]

[8 0 2]

[7 9 3 8 0 2]

Input number => 3

Size of Stack = 6

Input number => 4

Input value of element => 6

Input number => 2

[7 9 3 8 0 2 6]

Input number => 5

Element = 6 is deleted

Input number => 2

[7 9 3 8 0 2]

Input number => 6

Input number => 2

[0 2 3 7 8 9]

Input number => 7

Input number => 1

Input number of elements => 10

[4 8 3 9 0]

[5 2 2 7 3]

[4 8 3 9 0 5 2 2 7 3]

Input number => 6

Input number => 2

```
[ 0 2 2 3 3 4 5 7 8 9 ]
```

Input number => 0

```
MacBook-Pro-MacBook:~ macbookpro$ cat main1.cpp
#include "lab261.h"
```

```
int main(){
    Stack S, S1, S2; Init(S); Tvalue v; int p=1,n,k;
    printf("\nChoose action:"
        "\n1. Create Random Stack\n2. Print Stack\n3. Size of Stack"
        "\n4. Insert of Stack"
        "\n5. Delete from front of Stack"
        "\n6. Sorting Stack\n7. Clear Stack"
        "\n0. Exit");
    while(p){
        printf("\nInput number => "); scanf("%d",&p);
        if(p==1)
        { printf("Input number of elements => "); scanf("%d",&n);
          Init(S1);
          for(int i=0;i<n/2;i++){v=rand()%10; Push(S1,v);}
          Display(S1);
          Init(S2);
          for(int i=0;i<n-n/2;i++){v=rand()%10; Push(S2,v);}
          Display(S2); Cat(S1,S2); Append(S,S1); Display(S);}
        if(p==2) Display(S);
        if(p==3) { n=Size(S); printf("\nSize of Stack = %d\n",n); }
        if(p==4) { printf("\nInput value of element => "); scanf("%d",&v); Push(S,v);}
        if(p==5) { v=Pop(S); printf("\nElement = %d is deleted\n",v); }
        if(p==6) Sort(S);
        if(p==7) Init(S);
        if(p==0) break;}}
MacBook-Pro-MacBook:~ macbookpro$ cat lab261.h
```

```
#ifndef LAB261_H
#define LAB261_H
#include<stdio.h>
#include<stdlib.h>
#define N 100
#define Tvalue int

struct EL_St{ Tvalue V; EL_St *next;};
struct Stack{ EL_St *first;int size;};
```

```
void Init(Stack &S);
int Empty(Stack S);
void Push(Stack &S, Tvalue V);
Tvalue Pop(Stack &S);
Tvalue Top(Stack S);
int Size(Stack S);
void Cat(Stack &S1, Stack &S2);
void Append(Stack &S1, Stack &S2);
void Display(Stack S);
void Search_Swap(Stack &S, Stack &S1);
void Sort(Stack &S);
```

```
#endif
```

```
MacBook-Pro-MacBook:~ macbookpro$ cat lab261.cpp
#include "lab261.h"
```

```
void Init(Stack &S){ S.first=0; S.size=0;}
```

```
int Empty(Stack S){return S.first==0;}
```

```
void Push(Stack &S, Tvalue V){
    EL_St *t=new EL_St;
    t->next=S.first; S.first=t;
    S.first->V=V; S.size++;}
```

```
Tvalue Pop(Stack &S){
    if(Empty(S)) printf("STACK IS EMPTY");
    else { Tvalue V=S.first->V; S.first=S.first->next; S.size--; return V;}}
```

```
Tvalue Top(Stack S){
    if(Empty(S)) printf("STACK IS EMPTY");
    else return S.first->V;}
```

```

int Size(Stack S){return S.size;}

void Cat(Stack &S1, Stack &S2){
    Tvalue V;
    if(!Empty(S2)){ V=Pop(S2); Cat(S1,S2); Push(S1,V);}}

void Append(Stack &S1, Stack &S2){
    Tvalue V;
    if(Empty(S1)) Cat(S1,S2);
    else{ V=Pop(S1); Append(S1,S2); Push(S1,V);}}

void Display(Stack S){
    if(Empty(S)) printf("\nSTACK IS EMPTY\n");
    else{
        printf("\n["); El_St *t=S.first;
        while(t) { printf("%d ",t->V); t=t->next;}
        printf("]\n");}}

void Search_Swap(Stack &S, Stack &S1){
    bool swap=0; int f=Pop(S), s=Top(S1);
    while(f>s && !Empty(S1)){ s=Pop(S1); Push(S,s);
    if (!Empty(S1)) s=Top(S1);
    swap=1;}
    if(swap){ Push(S1,f); swap=0;}}

void Sort(Stack &S){
    Stack S1; Tvalue V,V1;
    Init(S1); int k=1;
    while (!Empty(S)){
        int e = Pop(S);
        if(Empty(S1) || Top(S1)>=e) Push(S1,e);
        else{ Push(S,e); Search_Swap(S,S1);}}
    while (!Empty(S1)) Push(S,Pop(S1));
    Init(S1);}
MacBook-Pro-MacBook:~ macbookpro$ g++ -o lab261 lab261.cpp main1.cpp
MacBook-Pro-MacBook:~ macbookpro$ ls |tail -7
lab261
lab261.cpp
lab261.h
main.cpp
main1.cpp
pslog_20200707_123036.log
zag.txt
MacBook-Pro-MacBook:~ macbookpro$ ./lab261

```

Choose action:

1. Create Random Stack
2. Print Stack
3. Size of Stack
4. Insert of Stack
5. Delete from front of Stack
6. Sorting Stack
7. Clear Stack
0. Exit

Input number => 1

Input number of elements => 8

[8 3 9 7]

[8 4 2 0]

[8 4 2 0 8 3 9 7]

Input number => 4

Input value of element => 8

Input number => 2

[8 8 4 2 0 8 3 9 7]

Input number => 3

Size of Stack = 9

```
Input number => 5
Element = 8 is deleted
Input number => 6
Input number => 2
[9 8 8 7 4 3 2 0 ]
Input number => 7
Input number => 1
Input number of elements => 15
[7 2 2 5 0 9 3 ]
[9 9 3 2 0 9 7 3 ]
[9 9 3 2 0 9 7 3 7 2 2 5 0 9 3 ]
Input number => 6
Input number => 2
[9 9 9 9 7 7 5 3 3 3 2 2 2 0 0 ]
Input number => 0
MacBook-Pro-MacBook:~ macbookpro$
```

9. Дневник отладки должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	Дом.	23.05.22	19:50	Вывод уведомления о пустом стеке при выполнении сортировки	Ошибка исправлена; более подробно рассмотрен алгоритм сортировки	В процедуре Search_Swap добавлена проверка на пустоту вспомогательного стека при вызове Top

10. Замечания автора по существу работы

Эта лабораторная работа очень полезна, она отлично развивает мышление и учит программированию на языке Си.

11. Выводы

Лабораторная работа №25: Я изучила принцип работы утилиты make. Лабораторная работа №26: Я составила и отладила модуль определений и модуль реализации по заданной схеме модуля определений для абстрактного (пользовательского) типа данных (стека, очереди, списка или дека, в зависимости от варианта задания). Я составила программный модуль, сортирующий экземпляры указанного абстрактного типа данных заданным методом, используя только операции, импортированные из модуля UUDT.

Недочёты при выполнении задания могут быть устранены следующим образом: больше практиковаться в написании программ на языке программирования Си.

Подпись студента Минеева С.А