Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 6

По дисциплине «Основы алгоритмизации программирования»

На тему «Динамические структуры данных. Односвязные списки»

Выполнил:

Студент 1 курса 6 группы

Кучерук Николай Петрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Задание**

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **9** | Создать список, содержащий элементы вещественного типа. Найти среднее значение отрицательных элементов. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <fstream>  using namespace std;  struct node//элемент односвязного списка  {  float data;//информация, хранящаяся в элементе  node\* next;//указатель на следующий элемент списка  node(float data) {//конструктор элемента  this->data = data;  this->next = NULL;  }  };  struct list//структура односвязного списка  {  node\* head,\*tail;//указатели на первый и последний элемент списка  list() {//конструктор списка  this->head = this->tail = NULL;  }  ~list() {//т.к список является динамичной структурой, то для предотвращения утечки памяти создаем деструктор  while (head != NULL) pop\_front();  }  void pop\_front() {//метод для удаления элемента в начале списка  if (head == NULL) return;  if (head == tail){  delete tail;  head = tail = NULL;  return;  }  node\* node = head;  head = node->next;  delete node;  }  void push\_front(float data) {//метод для добавления элемента в начало списка  node\* Node = new node(data);  Node->next = head;  head = Node;  if (tail == NULL)tail = Node;  }  void pop\_number(float number) {//метод для удаления элементов содержащих число на выбор пользователя  node\* Node = head;  node\* Node2 = Node;  while(Node!=NULL){  if (Node->data == number)  {  if (head == Node) { head = Node->next; }  if (Node->next == NULL) { tail = Node2; }  Node2->next = Node->next;  delete Node;  return;  }  Node = Node->next;  Node2->next = Node;  }    }  void output(){//метод для вывода списка  for (node\* Node = head; Node != NULL; Node = Node->next) {  cout << Node->data << " ";  }  cout << endl;  }  void solution() {//метод для получения ответа на поставленную задачу  float sum = 0.0;  float counter = 0.0;  for (node\* Node = head; Node != NULL; Node = Node->next) {  if (Node->data < 0) { sum += Node->data; counter++;}  }  cout << sum / counter<<endl;  }  };  void menu(void); //функция, показывающая меню  void filewr(list& ); //функция записи списка в файл  void filerd(list&); //функция получения списка из файла  int main()  {  setlocale(LC\_CTYPE, "Russian");  list first;//создаем список  int choice;  float value;  menu(); // вывести меню  cout << " ? ";  cin >> choice;  while (choice != 6)  {  switch (choice)  {  case 1: cout << "Введите число " << endl; // добавить число в список  cin >> value;  first.push\_front(value);  cout << "Cписок: ";  first.output();  break;  case 2:  {  cout << "Введите удаляемое число ";  cin >> value;  first.pop\_number(value);  cout << "Удалено число " << value << endl;  first.output();  }  break;  case 3: first.solution(); // вычисление суммы  break;  case 4:  {  filewr(first);  cout << "Файл записан " << endl;  break;  }  case 5:  {  filerd(first);  cout << "Данные из файла переписаны в список " << endl;  first.output();  break;  }  default: cout << "Неправильный выбор" << endl;  menu();  break;  }  cout << "? ";  cin >> choice;  }  cout << "Конец" << endl;  return 0;  }  void menu(void) //Вывод меню  {  cout << "Сделайте выбор:" << endl;  cout << " 1 - Ввод числа" << endl;  cout << " 2 - Удаление числа" << endl;  cout << " 3 - Вычисление среднего значения отрицательных чисел" << endl;  cout << " 4 - Ввод в файл" << endl;  cout << " 5 - Вывод из файла" << endl;  cout << " 6 - Выход" << endl;  }  void filewr(list &p) {//запись данных списка в файл  ofstream file("list.txt");  for (node\* Node = p.head; Node != NULL; Node = Node->next) {  file << Node->data << " ";  }  file.close();  }  void filerd(list& temp) {//чтение данных из файла в список  ifstream file("list.txt");  if (file.fail())  {  cout << "\n File cannot be opened";  exit(1);  }  while (temp.head != NULL) temp.pop\_front();  float a;  while (file>>a) {  temp.push\_front(a);  }  file.close();  } |
| **Результат программы** |
|  |

Дополнительные варианты

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **2** | Создать список, содержащий элементы вещественного типа. Найти среднее значение положительных элементов. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <fstream>  using namespace std;  struct node//элемент односвязного списка  {  float data;//информация, хранящаяся в элементе  node\* next;//указатель на следующий элемент списка  node(float data) {//конструктор элемента  this->data = data;  this->next = NULL;  }  };  struct list//структура односвязного списка  {  node\* head,\*tail;//указатели на первый и последний элемент списка  list() {//конструктор списка  this->head = this->tail = NULL;  }  ~list() {//т.к список является динамичной структурой, то для предотвращения утечки памяти создаем деконструктор  while (head != NULL) pop\_front();  }  void pop\_front() {//метод для удаления элемента в начале списка  if (head == NULL) return;  if (head == tail){  delete tail;  head = tail = NULL;  return;  }  node\* node = head;  head = node->next;  delete node;  }  void push\_front(float data) {//метод для добавления элемента в начало списка  node\* Node = new node(data);  Node->next = head;  head = Node;  if (tail == NULL)tail = Node;  }  void pop\_number(float number) {//метод для удаления элементов содержащих число на выбор пользователя  node\* Node = head;  node\* Node2 = Node;  while(Node!=NULL){  if (Node->data == number)  {  if (head == Node) { head = Node->next; }  if (Node->next == NULL) { tail = Node2; }  Node2->next = Node->next;  delete Node;  return;  }  Node = Node->next;  Node2->next = Node;  }    }  void output(){//метод для вывода списка  for (node\* Node = head; Node != NULL; Node = Node->next) {  cout << Node->data << " ";  }  cout << endl;  }  void solution() {//метод для получения ответа на поставленную задачу  float sum = 0.0;  float counter = 0.0;  for (node\* Node = head; Node != NULL; Node = Node->next) {  if (Node->data > 0) { sum += Node->data; counter++;}  }  cout << sum / counter<<endl;  }  };  void menu(void); //функция, показывающая меню  void filewr(list& ); //функция записи списка в файл  void filerd(list&); //функция получения списка из файла  int main()  {  setlocale(LC\_CTYPE, "Russian");  list first;//создаем список  int choice;  float value;  menu(); // вывести меню  cout << " ? ";  cin >> choice;  while (choice != 6)  {  switch (choice)  {  case 1: cout << "Введите число " << endl; // добавить число в список  cin >> value;  first.push\_front(value);  cout << "Cписок: ";  first.output();  break;  case 2:  {  cout << "Введите удаляемое число ";  cin >> value;  first.pop\_number(value);  cout << "Удалено число " << value << endl;  first.output();  }  break;  case 3: first.solution(); // вычисление суммы  break;  case 4:  {  filewr(first);  cout << "Файл записан " << endl;  break;  }  case 5:  {  filerd(first);  cout << "Данные из файла переписаны в список " << endl;  first.output();  break;  }  default: cout << "Неправильный выбор" << endl;  menu();  break;  }  cout << "? ";  cin >> choice;  }  cout << "Конец" << endl;  return 0;  }  void menu(void) //Вывод меню  {  cout << "Сделайте выбор:" << endl;  cout << " 1 - Ввод числа" << endl;  cout << " 2 - Удаление числа" << endl;  cout << " 3 - Вычисление среднего значения положительных чисел" << endl;  cout << " 4 - Ввод в файл" << endl;  cout << " 5 - Вывод из файла" << endl;  cout << " 6 - Выход" << endl;  }  void filewr(list &p) {//запись данных списка в файл  ofstream file("list.txt");  for (node\* Node = p.head; Node != NULL; Node = Node->next) {  file << Node->data << " ";  }  file.close();  }  void filerd(list& temp) {//чтение данных из файла в список  ifstream file("list.txt");  if (file.fail())  {  cout << "\n File cannot be opened";  exit(1);  }  while (temp.head != NULL) temp.pop\_front();  float a;  while (file>>a) {  temp.push\_front(a);  }  file.close();  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **12** | Создать список, содержащий элементы вещественного типа. Найти среднее значение положительных элементов. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <fstream>  using namespace std;  struct node//элемент односвязного списка  {  float data;//информация, хранящаяся в элементе  node\* next;//указатель на следующий элемент списка  node(float data) {//конструктор элемента  this->data = data;  this->next = NULL;  }  };  struct list//структура односвязного списка  {  node\* head,\*tail;//указатели на первый и последний элемент списка  list() {//конструктор списка  this->head = this->tail = NULL;  }  ~list() {//т.к список является динамичной структурой, то для предотвращения утечки памяти создаем деструктор  while (head != NULL) pop\_front();  }  void pop\_front() {//метод для удаления элемента в начале списка  if (head == NULL) return;  if (head == tail){  delete tail;  head = tail = NULL;  return;  }  node\* node = head;  head = node->next;  delete node;  }  void push\_front(float data) {//метод для добавления элемента в начало списка  node\* Node = new node(data);  Node->next = head;  head = Node;  if (tail == NULL)tail = Node;  }  void pop\_number(float number) {//метод для удаления элементов содержащих число на выбор пользователя  node\* Node = head;  node\* Node2 = Node;  while(Node!=NULL){  if (Node->data == number)  {  if (head == Node) { head = Node->next; }  if (Node->next == NULL) { tail = Node2; }  Node2->next = Node->next;  delete Node;  return;  }  Node = Node->next;  Node2->next = Node;  }    }  void output(){//метод для вывода списка  for (node\* Node = head; Node != NULL; Node = Node->next) {  cout << Node->data << " ";  }  cout << endl;  }  void solution() {//метод для получения ответа на поставленную задачу  float sum = 0.0;  float counter = 0.0;  for (node\* Node = head; Node != NULL; Node = Node->next) {  if (Node->data > 0) { sum += Node->data; counter++;}  }  cout << sum / counter<<endl;  }  };  void menu(void); //функция, показывающая меню  void filewr(list& ); //функция записи списка в файл  void filerd(list&); //функция получения списка из файла  int main()  {  setlocale(LC\_CTYPE, "Russian");  list first;//создаем список  int choice;  float value;  menu(); // вывести меню  cout << " ? ";  cin >> choice;  while (choice != 6)  {  switch (choice)  {  case 1: cout << "Введите число " << endl; // добавить число в список  cin >> value;  first.push\_front(value);  cout << "Cписок: ";  first.output();  break;  case 2:  {  cout << "Введите удаляемое число ";  cin >> value;  first.pop\_number(value);  cout << "Удалено число " << value << endl;  first.output();  }  break;  case 3: first.solution(); // вычисление суммы  break;  case 4:  {  filewr(first);  cout << "Файл записан " << endl;  break;  }  case 5:  {  filerd(first);  cout << "Данные из файла переписаны в список " << endl;  first.output();  break;  }  default: cout << "Неправильный выбор" << endl;  menu();  break;  }  cout << "? ";  cin >> choice;  }  cout << "Конец" << endl;  return 0; |
| **Результат программы** |
|  |

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **16** | Создать список, содержащий элементы вещественного типа. Найти сумму элементов, каждый из которых меньше числа 9. |

|  |  |
| --- | --- |
| **Код программы** |  |
| #include <iostream>  #include <fstream>  using namespace std;  struct node//элемент односвязного списка  {  float data;//информация, хранящаяся в элементе  node\* next;//указатель на следующий элемент списка  node(float data) {//конструктор элемента  this->data = data;  this->next = NULL;  }  };  struct list//структура односвязного списка  {  node\* head,\*tail;//указатели на первый и последний элемент списка  list() {//конструктор списка  this->head = this->tail = NULL;  }  ~list() {//т.к список является динамичной структурой, то для предотвращения утечки памяти создаем деструктор  while (head != NULL) pop\_front();  }  void pop\_front() {//метод для удаления элемента в начале списка  if (head == NULL) return;  if (head == tail){  delete tail;  head = tail = NULL;  return;  }  node\* node = head;  head = node->next;  delete node;  }  void push\_front(float data) {//метод для добавления элемента в начало списка  node\* Node = new node(data);  Node->next = head;  head = Node;  if (tail == NULL)tail = Node;  }  void pop\_number(float number) {//метод для удаления элементов содержащих число на выбор пользователя  node\* Node = head;  node\* Node2 = Node;  while(Node!=NULL){  if (Node->data == number)  {  if (head == Node) { head = Node->next; }  if (Node->next == NULL) { tail = Node2; }  Node2->next = Node->next;  delete Node;  return;  }  Node = Node->next;  Node2->next = Node;  }    }  void output(){//метод для вывода списка  for (node\* Node = head; Node != NULL; Node = Node->next) {  cout << Node->data << " ";  }  cout << endl;  }  void solution() {//метод для получения ответа на поставленную задачу  float sum = 0.0;  float counter = 0.0;  for (node\* Node = head; Node != NULL; Node = Node->next) {  if (Node->data < 9) { sum += Node->data; }  }  cout << sum <<endl;  }  };  void menu(void); //функция, показывающая меню  void filewr(list& ); //функция записи списка в файл  void filerd(list&); //функция получения списка из файла  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  list first;//создаем список  int choice;  float value;  menu(); // вывести меню  cout << " ? ";  cin >> choice;  while (choice != 6)  {  switch (choice)  {  case 1: cout << "Введите число " << endl; // добавить число в список  cin >> value;  first.push\_front(value);  cout << "Cписок: ";  first.output();  break;  case 2:  {  cout << "Введите удаляемое число ";  cin >> value;  first.pop\_number(value);  cout << "Удалено число " << value << endl;  first.output();  }  break;  case 3: first.solution(); // вычисление суммы  break;  case 4:  {  filewr(first);  cout << "Файл записан " << endl;  break;  }  case 5:  {  filerd(first);  cout << "Данные из файла переписаны в список " << endl;  first.output();  break;  }  default: cout << "Неправильный выбор" << endl;  menu();  break;  }  cout << "? ";  cin >> choice;  }  cout << "Конец" << endl;  return 0;  }  void menu(void) //Вывод меню  {  cout << "Сделайте выбор:" << endl;  cout << " 1 - Ввод числа" << endl;  cout << " 2 - Удаление числа" << endl;  cout << " 3 - Вычисление суммы чисел больше 9" << endl;  cout << " 4 - Ввод в файл" << endl;  cout << " 5 - Вывод из файла" << endl;  cout << " 6 - Выход" << endl;  }  void filewr(list &p) {//запись данных списка в файл  ofstream file("list.txt");  for (node\* Node = p.head; Node != NULL; Node = Node->next) {  file << Node->data << " ";  }  file.close();  }  void filerd(list& temp) {//чтение данных из файла в список  ifstream file("list.txt");  if (file.fail())  {  cout << "\n File cannot be opened";  exit(1);  }  while (temp.head != NULL) temp.pop\_front();  float a;  while (file>>a) {  temp.push\_front(a);  }  file.close();  } |
| **Результат программы** |
|  |