**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное

учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники

Отделение автоматизации и робототехники

Направление мехатроника и робототехника

Отчет

по лабораторной работе №5

по дисциплине

**«**Основы программирования и алгоритмизации**»**

**Структуры в C++**

Выполнил:

Студент группы 8E21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.С. Моисеев

Проверил:

Ассистент ОАР ИШИТР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Я.О. Кургинов

Томск 2023

Вариант 10

Цель работы:

Поработать со структурами в C++.

# Задание 1

При заказе в KFC формируется электронная очередь. Если составляющие заказа готовы, то заказ собирается и отдается покупателю. Если нет, то пропускается и собирается следующий.

Реализовать структуру данных, представляющую электронную очередь.

Алгоритмы должны выполнять добавление в очередь и удаление из очереди при получении заказа, выбор заказа, который собирается на данный момент, исходя из порядка в очереди и готовности позиций заказа. Учесть что кухня периодически пополняет запас готовых позиций.

**Блок схема** (Рисунок 1)

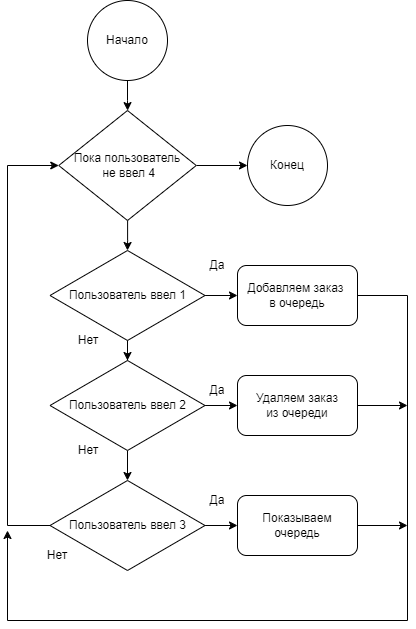


Рисунок 1 - Блок-схема к заданию 1.

Текст программы (Листинг 1, Приложение А)

**Результаты работы** (Рисунок 2)

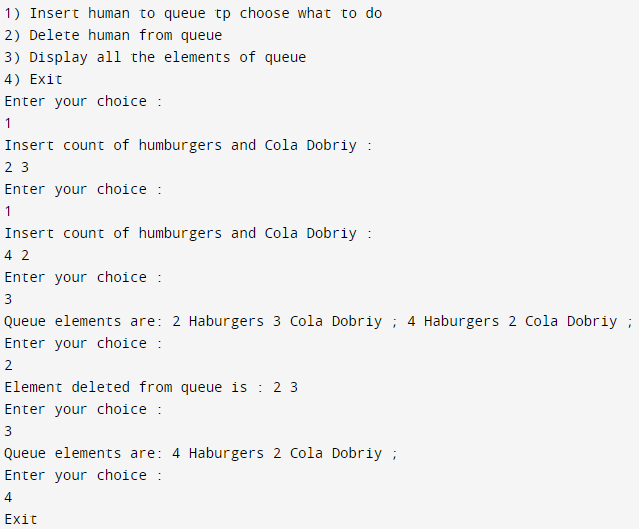
****

Рисунок 2 - Скриншот результата работы программы к заданию 1 на консоли.

# Вывод:

В результате лабораторной работы были решены поставленные задачи со структурами, используя C++.

**Приложение А**

Листинг 1 – Программа к заданию 1.

|  |
| --- |
| 1. #include <iostream> 2. using namespace std; 3. struct node { 4. int data; 5. int data1; 6. struct node \*next; 7. }; 8. struct node\* front = NULL; 9. struct node\* rear = NULL; 10. struct node\* temp; 11. struct queue{ 12. int kfccombo; 13. int cola; 14. }; 15. void Insert() { 16. int val; 17. int val1; 18. cout<<"Insert count of humburgers and Cola Dobriy : "<<endl; 19. cin>>val; 20. cin>>val1; 21. if (rear == NULL) { 22. rear = (struct node \*)malloc(sizeof(struct node)); 23. rear->next = NULL; 24. rear->data = val; 25. rear->data1 = val1; 26. front = rear; 27. } else { 28. temp=(struct node \*)malloc(sizeof(struct node)); 29. rear->next = temp; 30. temp->data = val; 31. temp->data1 = val1; 32. temp->next = NULL; 33. rear = temp; 34. } 35. } 36. void Delete() { 37. temp = front; 38. if (front == NULL) { 39. cout<<"Underflow"<<endl; 40. return; 41. } 42. else 43. if (temp->next != NULL) { 44. temp = temp->next; 45. cout<<"Element deleted from queue is : "<<front->data << " " << front->data1<<endl; 46. free(front); 47. front = temp; 48. } else { 49. cout<<"Element deleted from queue is : "<<front->data << " " << front->data1 <<endl; 50. free(front); 51. front = NULL; 52. rear = NULL; 53. } 54. } 55. void Display() { 56. temp = front; 57. if ((front == NULL) && (rear == NULL)) { 58. cout<<"Queue is empty"<<endl; 59. return; 60. } 61. cout<<"Queue elements are: "; 62. while (temp != NULL) { 63. int x = temp->data; 64. if(x!=0) cout << temp->data << " Haburgers" << " "; 65. if(x!=0) cout << temp->data1 << " Cola Dobriy" << " "; 66. cout << "; "; 67. temp = temp->next; 68. } 69. cout<<endl; 70. } 71. int main() { 72. int ch; 73. cout<<"1) Insert human to queue tp choose what to do"<<endl; 74. cout<<"2) Delete human from queue"<<endl; 75. cout<<"3) Display all the elements of queue"<<endl; 76. cout<<"4) Exit"<<endl; 77. do { 78. cout<<"Enter your choice : "<<endl; 79. cin>>ch; 80. switch (ch) { 81. case 1: Insert(); 82. break; 83. case 2: Delete(); 84. break; 85. case 3: Display(); 86. break; 87. case 4: cout<<"Exit"<<endl; 88. break; 89. default: cout<<"Invalid choice"<<endl; 90. } 91. } while(ch!=4); 92. return 0; 93. } |