Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №6 з дисципліни

«Алгоритми структури даних»

«Дослідження рекурсивних алгоритмів »

Варіант 34

Виконав студент ІП-1134 Шамков Іван Дмитрович

( прізвище, ім'я, по батькові)

Перевірив викладач Мартинова Оксана Петрівна

( прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота №6

Дослідження рекурсивних алгоритмів

Лабораторна робота 6

Дослідження рекурсивних алгоритмів

*Мета* – дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

Варіант: 34

*Умова задачі:*



*Математична модель:*

| Змінна | Тип | Ім’я | Призначення |
| --- | --- | --- | --- |
| Число у 16 системі числення | Рядковий | num | Початкове дане |
| Номер символу у числі | Цілий | i | Проміжне значення |
| Номер цифри, яку будемо використовувати | Цілий | j | Проміжне значення |
| Цифра у десятковій системі числення | Цілий | ch | Проміжне значення |
| Запис цифри у 16 системі числення | Символьний | ch1 | Проміжне значення |
| Цифра у десятковій системі числення | Цілий | char\_found | Проміжне значення |
| Номер символу у системі ASCII, з якого починаємо відлік | Цілий | start | Проміжне значення |
| кількість збільшень відносно змінної start | Цілий | count | Проміжне значення |
| Число у десятковій системі числення | Цілий | res | Результат |

Постановка задачі:

Отже, математичне формулювання нашої задачі полягає в тому, щоб отримати число у шістнадцятковій системі числення, яке записуємо символами, адже в цій системі числення наявні цифри, які відображаємо як літери: A, B, C, D, E, F. Після цього викликаємо функцію ConvertToTen(), яка переводить число з шістнадцяткової системи числення у десяткову за формулою res+=ch\*pow(16, i).

Але для цього їй потрібно отримати значення ch. Це відбувається за допомогою виклику функції TakeChar(), яка вирізає з рядка num перше значення та перевіряє номер його коду за таблицею ASCII. Якщо він знаходиться у проміжку [48; 57], то викликаємо третю функцію FindID() та передаємо один набір значень, а якщо проміжку - [65; 70], то передаємо у функцію трохи інший набір значень. Ця остання функція вже і знаходить те, чому дорівнює обрана цифра з рядка num у десятковій системі. Повертаємо його назад до ConvertToTen(), обраховуємо значення res за формулою та викликаємо ConvertToTen() ще раз, але передаємо в неї коротший рядок num, збільшене на 1 j та зменшене на 1 i.

Вважатимемо pow(A, B) піднесенням числа А до степення В, а

С.size() - передача довжини рядка С.

Наступні функції є створеними власноруч:

ConvertToTen(string num, int i, int j=0, int res=0)

TakeChar(string num, int j)

FindID(int start, char ch1)

*Псевдокод:*

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо значення num

Крок 3. Переводимо число у десяткову систему числення

**Крок 1:**

Start

Деталізуємо значення num

Деталізуємо переведення числа у десяткову систему числення

End

**Крок 2:**

Start

Input num

Деталізуємо переведення числа у десяткову систему числення

End

**Крок 3:**

Start

Input num

i=num.size()-1

**ConvertToTen(num, i)**

End

Підпрограми

**ConvertToTen(num, i, j=0, res=0)**

**if** i == (-1)

then

**output** “ Your number in 10 is ”, res

**return**

**end if**

ch=**TakeChar(num, j)**

res += ch\*pow(16, i)

num[j] = char(32)

**ConvertToTen(num, i+1, j+1, res)**

**TakeChar( num, j)**

char ch1 = num[j]

**if** (48 <= int(ch1) **and** int(ch1) <= 57)

then

char\_found = **FindID(48, ch1)**

**return** char\_found;

**else if** (65 <= int(ch1) **and** int(ch1) <= 70)

then

char\_found = **FindID(65, ch1)**

**return** char\_found;

**end if**

**FindID(start, ch1)**

**if** (start == 65)

then

count=10

**else**

then

count=0

**end if**

**while** ( int(ch1) != start)

**repeat**

start+=1

count+=1

**end repeat**

**return** count;

*Блок схема:*

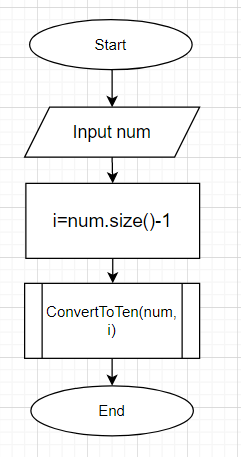
Крок 1



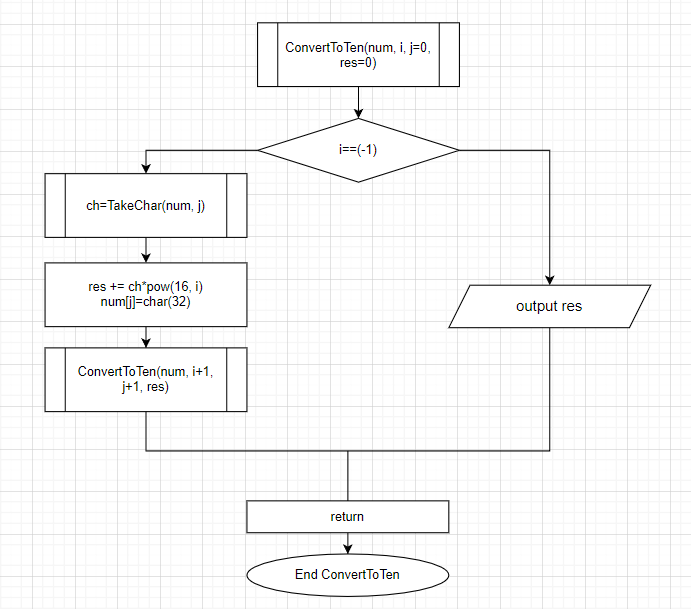
Крок 2

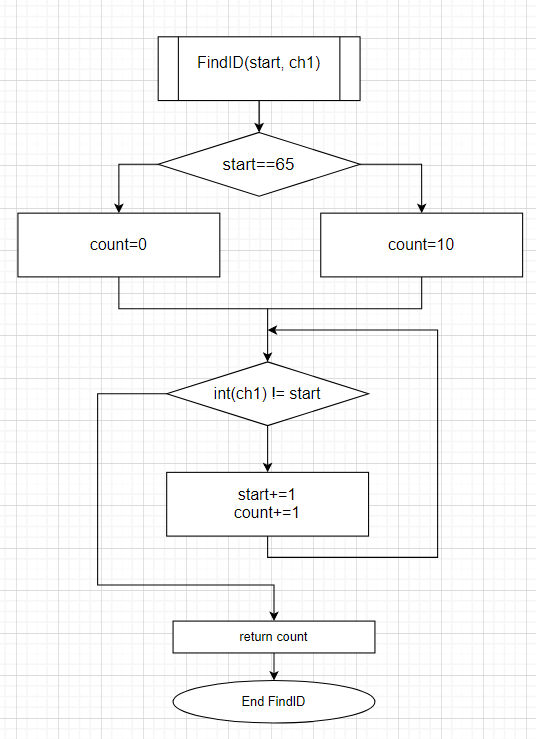
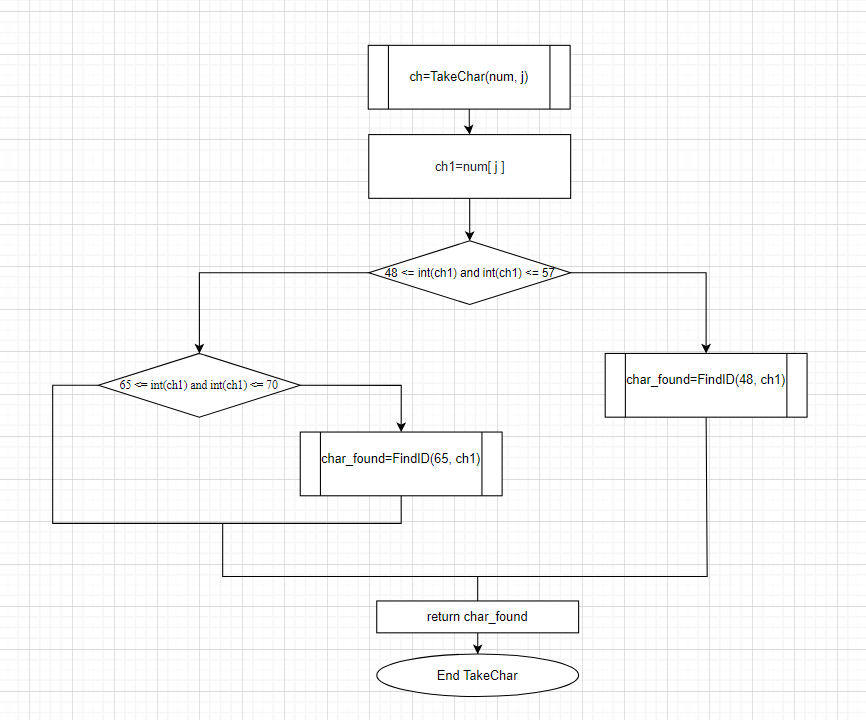


Крок 3



Підпрограми

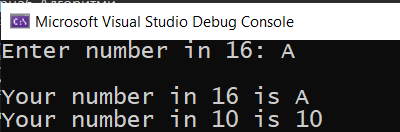


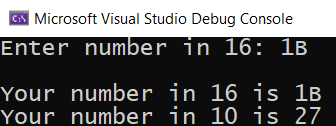
**

*Код на С++:*

**

*Копії екранних форм*

**

**

*Випробування алгоритму*

| Блок | Дія |
| --- | --- |
|  | Початок |
| 1 | Введення num= “A” |
| 2 | i=num.size()-1 = 0 |
| 3 | ConvertToTen(“А”, 0)  if і==(-1)(False)  сh=TakeChar(“A”, j=0)  res=0+10\*pow(16, 0)=10  num[0]=char(32) // num=” ”  ConvertToTen(“ ”, -1) |
| 4 | TakeChar(“A”, j=0)  ch1=num[0]= “A”  (48 <= int(ch1) and int(ch1) <= 57) (False)  (65 <= int(ch1) and int(ch1) <= 70)(True)  char\_found = FindID(65, ch1) |
| 5 | FindID(65, ch1)  start==65(True)  count=10  while int(“A”)!=start //65!=65 (False)  return count |
| 6 | ConvertToTen(“ ”, -1)  i==(-1) (True)  output(res) (10)  return |
|  | Кінець |

*Висновок*

Отже, виконавши цю лабораторну роботу, ми навчилися використовувати рекурсивні алгоритми. Їхня реалізація відбувається через підпрограми, які викликають самі себе. У таких підпрограмах головне врахувати крайній випадок, після якого закінчиться самовиклик, адже в іншому випадку ми можемо потрапити у нескінченний цикл. У процесі виконання ми сформулювали задачу, побудували математичну модель та псевдокод алгоритму, що допомогло нам краще її зрозуміти. Основною частиною алгоритму є переведення числа із шістнадцяткової у десяткову систему. Це реалізовано за допомогою рекурсивної підпрограми під назвою ConvertToTen().Також їй “допомагають” такі дві підпрограми як TakeChar() та FindID(). Алгоритм працює за принципом перетворення символьного значення однієї із цифр числа у шістнадцятковій системі в десяткову. Наприклад, ми переводимо символ “8” у число 8, а символ “F” у число 15. Після цього ми використовуємо формулу для переводу. Крайнім випадком, коли рекурсивна підпрограма перестає себе викликати є ситуація, коли i=-1, тобто ми вже перевели усі цифри з числа у шістнадцятковій системі в десяткову. У такому разі виводимо результат res та закінчуємо роботу підпрограм.