

**МIНIСТЕРСТВО  ОСВIТИ І НАУКИ  УКРАЇНИ**

**НАЦІОНАЛЬНИЙ   ТЕХНІЧНИЙ УНІВЕРСИТЕТ   УКРАЇНИ**

“**КИЇВСЬКИЙ  ПОЛІТЕХНІЧНИЙ  ІНСТИТУТ**

**ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

**Факультет прикладної математики**

**Кафедра програмного забезпечення комп’ютерних систем**

**Лабораторна робота № 5**

**з дисципліни “ Основи програмування ”**

**тема “ Символи та рядки ”**

|  |  |  |
| --- | --- | --- |
| **Виконала**  **студентка I курсу**  **групи КП-83**  **Снітко Маріанна Дмитрівна**  **(прізвище, ім’я, по батькові)**  **варіант № 22** |  | **Перевірив**  “**\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.**  **викладач**  **Гадиняк Руслан Анатолійович**  **(прізвище, ім’я, по батькові)** |

**Київ 2018**

**Мета роботи**

Навчитися працювати зі масивами символів та рядками.  
Вміти визначати класи символів та використовувати функції для роботи із символами.  
Навчитись використовувати стандартні функції для роботи з рядками для вирішення прикладних задач.  
Навчитися оформлювати консольну програму для зручності роботи користувача.

**Постановка завдання**

При старті консольної програми користувачу вивести меню, наприклад:

1. Characters

2. String

3. Quit

#### **Підменю 1: Символи**

У даному підменю у користувача має бути вибір одного із графічних класів символів, наприклад:

1. Alphanumeric

2. Alphabetic (lowercase)

3. Alphabetic (uppercase)

4. Alphabetic (all)

5. Decimal digit

6. Hexadecimal digit

7. Punctuation

0. < Back

При виборі одного із класу символів у меню (використовуючи цикл) **програмно** вивести через пробіл всі ASCII символи цього класу.

#### **Підменю 2: Рядок**

Перед переходом у дане меню користувачу необхідно ввести значення N - ціле додатнє число.  
У цьому меню у користувача має бути буфер символів (масив розміром N) для зберігання рядка.  
У початковому стані буфер повинен містити рядок із випадкових видимих символів (наприклад, j.g65&$$%@IUFjg?/).

Доступні операції над рядком:

1. Замінити рядок у масиві на введене рядкове значення із консолі (рядок може містити введені пробіли, використовувати fgets().
2. Очистити рядок.
3. Вивести підрядок із заданої позиції і заданої довжини.
4. Вивести список підрядків, розділених заданим символом.
5. Вивести найдовше слово.
6. Знайти та вивести всі цілі числа, що містяться у рядку.
7. Знайти та вивести суму всіх дробових чисел, що містяться у рядку.

Всі виводи рядків виконувати за допомогою стандартних функцій (puts,printf). Для цього варто створювати тимчасові масиви, у які розміщувати рядок, який потрібно вивести.

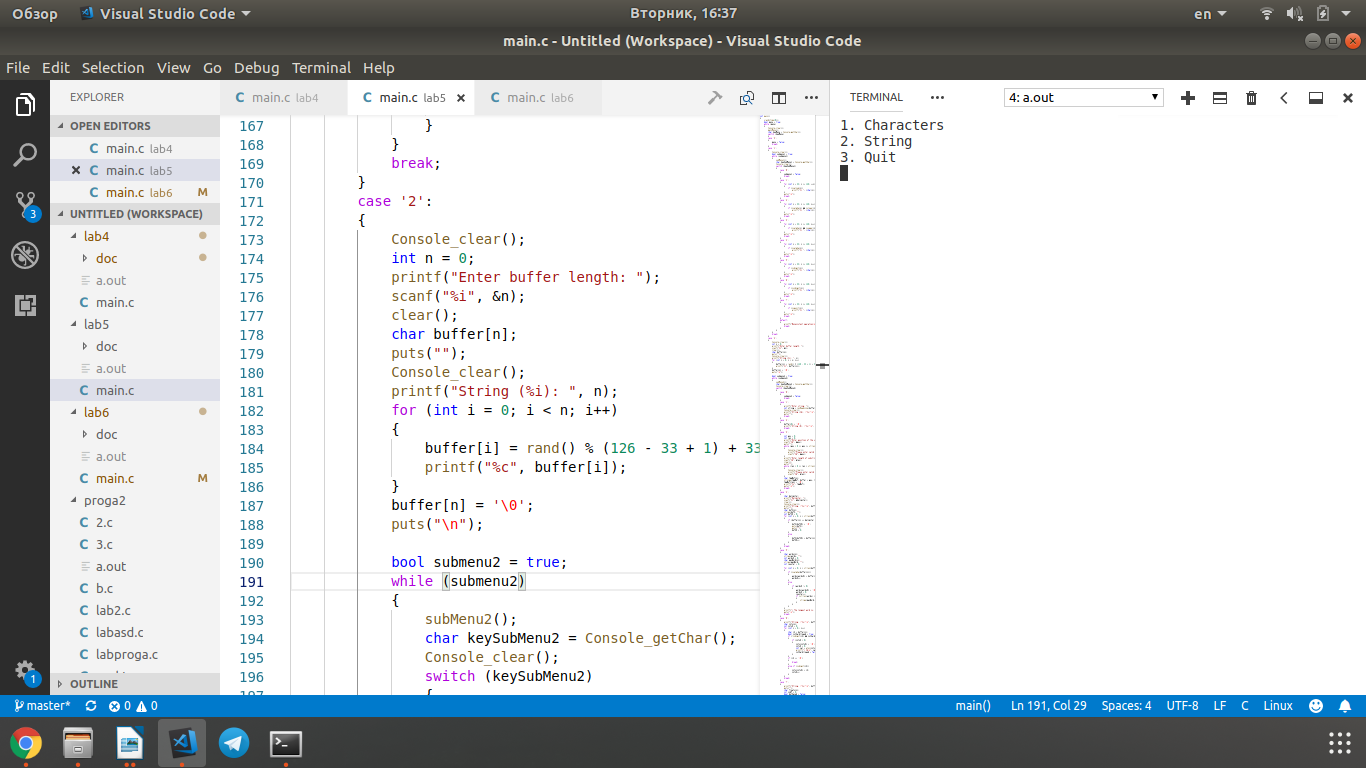
Отримання рядків від користувача виконувати за допомогою спеціальних функцій. Зчитувати рядок у циклі посимвольно не можна.

**Тексти коду програм**

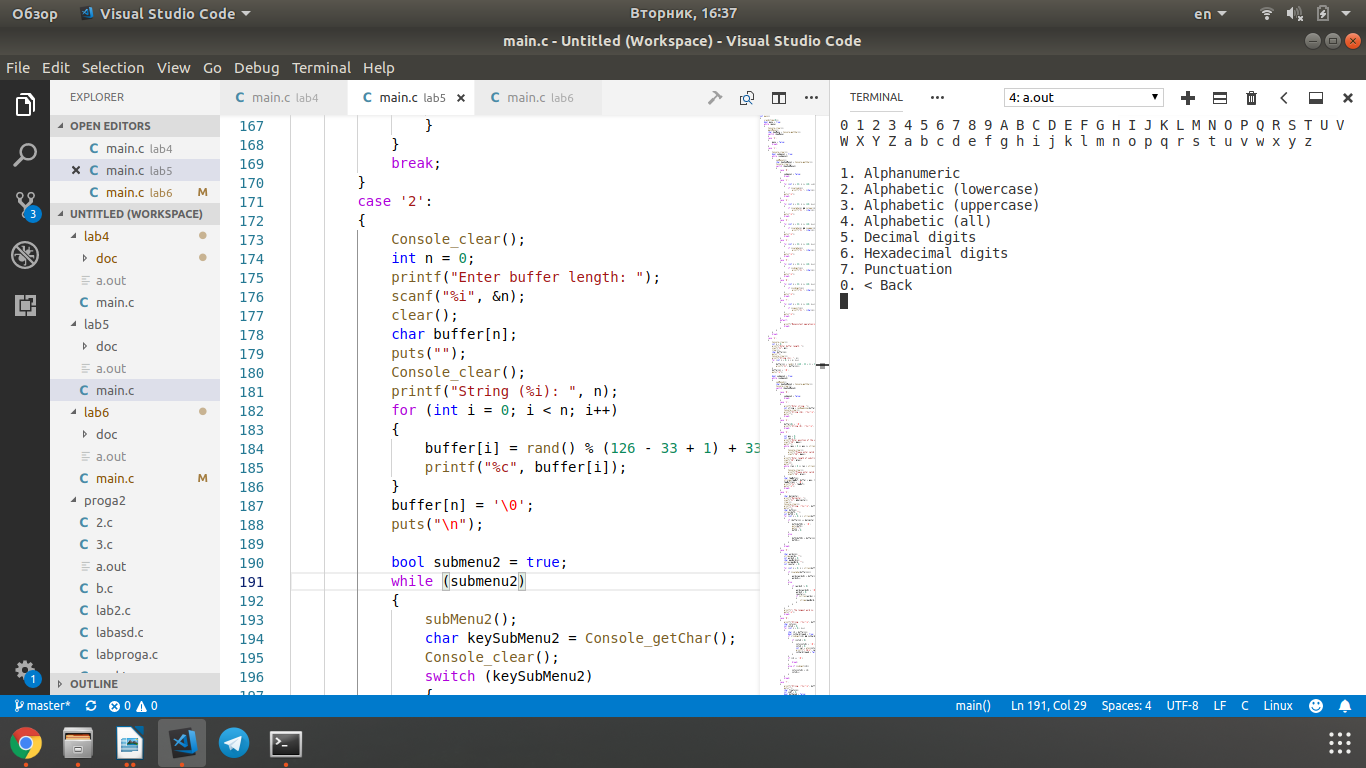
|  |
| --- |
| **main.c** |
| #include <stdio.h>  #include <math.h>  #include <stdlib.h>  #include <progbase.h>  #include <stdbool.h>  #include <progbase/console.h>  #include <ctype.h>  #include <string.h>  #include <time.h>  int conReadLine(char str[], int maxBufLen)  {  fgets(str, maxBufLen, stdin);  int bufLength = strlen(str);  if (str[bufLength - 1] == '\n')  {  str[bufLength - 1] = '\0';  bufLength -= 1;  }  else  {  for (char ch; (ch = getchar()) != '\n';)  {  }  }  return bufLength;  }  void mainMenu(void)  {  printf("1. Characters\n2. String\n3. Quit\n");  }  void subMenu1(void)  {  printf("1. Alphanumeric\n");  printf("2. Alphabetic (lowercase)\n");  printf("3. Alphabetic (uppercase)\n");  printf("4. Alphabetic (all)\n");  printf("5. Decimal digits\n");  printf("6. Hexadecimal digits\n");  printf("7. Punctuation\n");  printf("0. < Back\n");  }  void subMenu2(void)  {  printf("1. Fill the string\n");  printf("2. Reset string\n");  printf("3. Substring\n");  printf("4. List of substrings, divided by entered symbol\n");  printf("5. The longest word\n");  printf("6. Integers of string\n");  printf("7. Sum of all floats\n");  printf("0. < Back\n");  }  void clear()  {  while (getchar() != '\n')  ;  }  int main()  {  srand(time(0));  bool menu = true;  while (menu)  {  Console\_clear();  mainMenu();  char keyMenu = Console\_getChar();  switch (keyMenu)  {  case '3':  {  menu = false;  break;  }  case '1':  {  Console\_clear();  bool submenu1 = true;  while (submenu1)  {  subMenu1();  char keySubMenu1 = Console\_getChar();  Console\_clear();  switch (keySubMenu1)  {  case '0':  {  submenu1 = false;  break;  }  case '1':  {  for (int i = 33; i <= 126; ++i)  {  if (isalnum(i))  printf("%c ", (char)i);  }  puts("\n");  break;  }  case '2':  {  for (int i = 33; i <= 126; i++)  {  if (isalpha(i) && islower(i))  printf("%c ", (char)i);  }  puts("\n");  break;  }  case '3':  {  for (int i = 33; i <= 126; i++)  {  if (isalpha(i) && isupper(i))  printf("%c ", (char)i);  }  puts("\n");  break;  }  case '4':  {  for (int i = 33; i <= 126; i++)  {  if (isalpha(i))  printf("%c ", (char)i);  }  puts("\n");  break;  }  case '5':  {  for (int i = 33; i <= 126; i++)  {  if (isdigit(i))  printf("%c ", (char)i);  }  puts("\n");  break;  }  case '6':  {  for (int i = 33; i <= 126; i++)  {  if (isxdigit(i))  printf("%c ", (char)i);  }  puts("\n");  break;  }  case '7':  {  for (int i = 33; i <= 126; i++)  {  if (ispunct(i))  printf("%c ", (char)i);  }  puts("\n");  break;  }  default:  {  printf("Nonexistent operation\n");  break;  }  }  }  break;  }  case '2':  {  Console\_clear();  int n = 0;  printf("Enter buffer length: ");  scanf("%i", &n);  clear();  char buffer[n];  puts("");  Console\_clear();  printf("String (%i): ", n);  for (int i = 0; i < n; i++)  {  buffer[i] = rand() % (126 - 33 + 1) + 33;  printf("%c", buffer[i]);  }  buffer[n] = '\0';  puts("\n");  bool submenu2 = true;  while (submenu2)  {  subMenu2();  char keySubMenu2 = Console\_getChar();  Console\_clear();  switch (keySubMenu2)  {  case '0':  {  submenu2 = false;  break;  }  case '1':  {  printf("Enter string: ");  int string = conReadLine(buffer, n);  Console\_clear();  printf("String (%d): \"%s\"\n", string, buffer);  puts("");  break;  }  case '2':  {  buffer[0] = '\0';  printf("String (0): \"%s\"\n", buffer);  break;  }  case '3':  {  int pos = 0;  int len = 0;  printf("Enter position of the start of substring: ");  scanf("%d", &pos);  puts("");  while (pos < 0 || pos >= strlen(buffer))  {  Console\_clear();  printf("Please enter valid position\n");  scanf("%d", &pos);  }  printf("Enter length of substring: ");  scanf("%d", &len);  clear();  puts("");  while (len < 0 || len > strlen(buffer) - pos)  {  Console\_clear();  printf("Please enter valid length\n");  scanf("%d", &len);  }  char tmpBuf[n];  strncpy(tmpBuf, buffer + pos, len);  tmpBuf[len] = '\0';  printf("%s", tmpBuf);  puts("\n");  break;  }  case '4':  {  char delimiter;  printf("Delimiter: ");  scanf("%c", &delimiter);  clear();  Console\_clear();  printf("String: \"%s\"\n", buffer);  puts("");  char buf4[n];  strcpy(buf4, "");  int buf4X = 0;  for (int i = 0; i < strlen(buffer) + 1; i++)  {  if (buffer[i] == delimiter || buffer[i] == '\0')  {  buf4[buf4X] = '\0';  puts(buf4);  puts("");  buf4X = 0;  }  else  {  buf4[buf4X] = buffer[i];  buf4X++;  }  }  break;  }  case '5':  {  char words[n];  strcpy(words, "");  int wordsX = 0;  char maxWord[n];  strcpy(maxWord, "");  int counter = 0;  for (int i = 0; i < strlen(buffer) + 1; i++)  {  if (isalpha(buffer[i]))  {  words[wordsX] = buffer[i];  wordsX++;  }  else  {  if (wordsX != 0)  {  words[wordsX] = '\0';  wordsX = 0;  counter++;  if (strlen(words) > strlen(maxWord))  {  strcpy(maxWord, words);  }  }  }  }  printf("> The longest word is: %s", maxWord);  puts("\n");  break;  }  case '6':  {  printf("String: \"%s\"\n", buffer);  char ints[n];  int intsX = 0;  for (int i = 0;; i++)  {  char ch = buffer[i];  bool isCharAllowed = true;  if (!isdigit(ch) && isCharAllowed == true)  {  if (intsX > 0)  {  ints[intsX] = '\0';  intsX = 0;  int val = atoi(ints);  printf("> %d\n", val);  isCharAllowed = false;  }  }  if (ch == '\0')  {  break;  }  else if (isdigit(ch))  {  ints[intsX] = ch;  intsX++;  }  }  break;  }  case '7':  {  printf("String: \"%s\"\n", buffer);  puts("");  char floats[n];  int indF = 0;  bool dotSaved = false;  float val3 = 0;  for (int i = 0;; i++)  {  char ch3 = buffer[i];  if ((!isdigit(ch3) && (ch3 != '.')) || ((ch3 == '.') && dotSaved) || (ch3 == '\0'))  {  if (indF > 0)  {  if (indF > 1 && dotSaved)  {  floats[indF] = '\0';  val3 += atof(floats);  indF = 0;  dotSaved = false;  }  else  {  indF = 0;  dotSaved = false;  }  }  }  if (ch3 == '\0')  {  break;  }  else  {  if (isdigit(ch3) || ch3 == '.')  {  floats[indF] = ch3;  if (ch3 == '.')  {  dotSaved = true;  }  indF++;  }  }  }  printf("> Sum = %.2f", val3);  puts("\n");  break;  }  default:  {  printf("Nonexistent operation\n");  break;  }  }  }  break;  }  default:  {  break;  }  }  }  puts("Bye!");  return 0;  } |

**Приклади результатів**

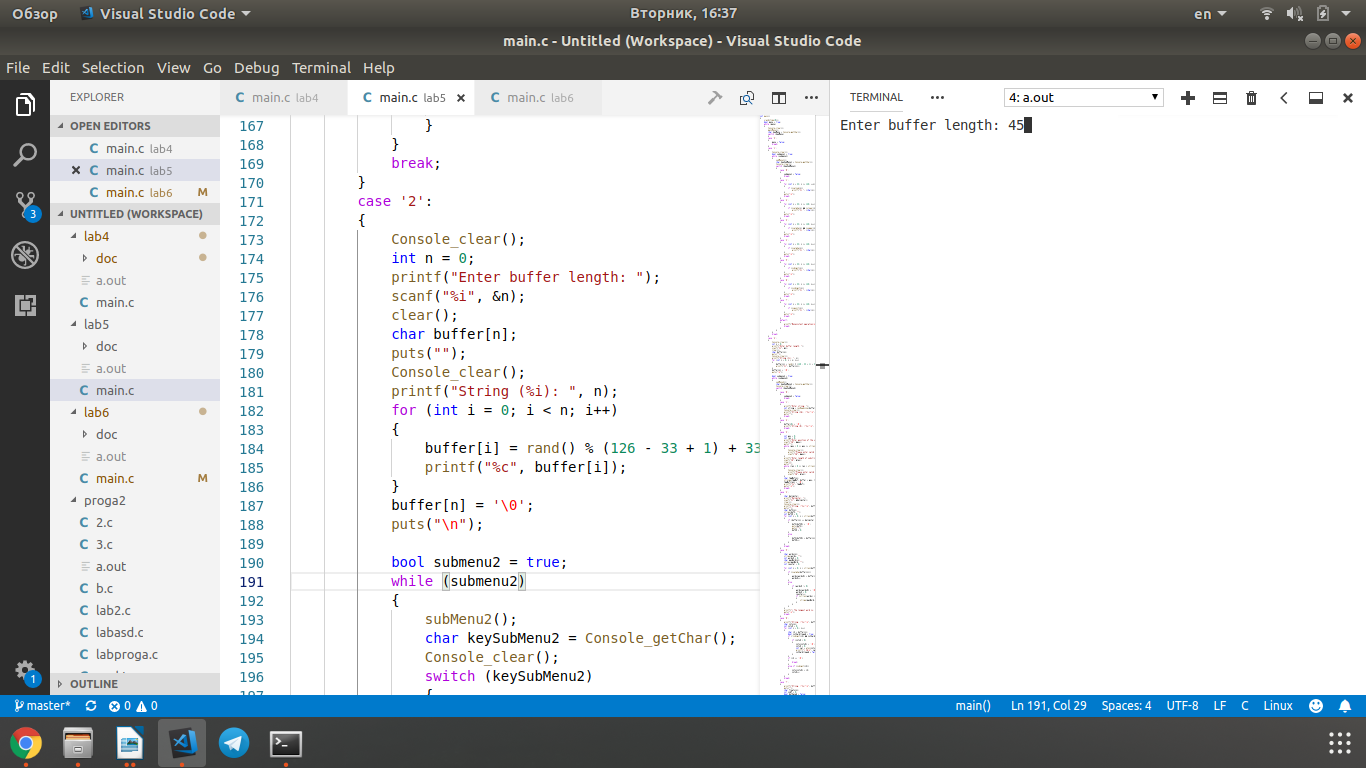
При старті програми у консолі виводиться головне меню:



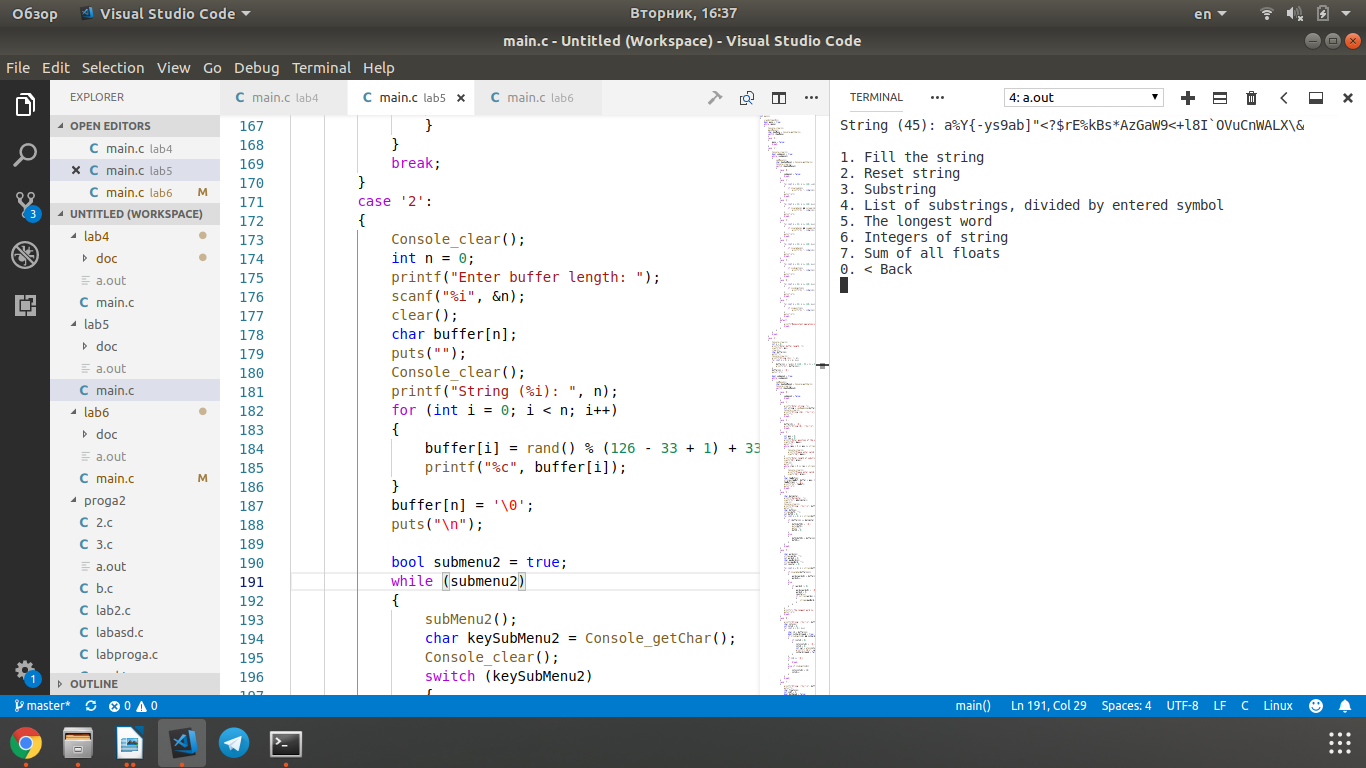
При вході у перше підменю пропонується обрати один з класів символів:



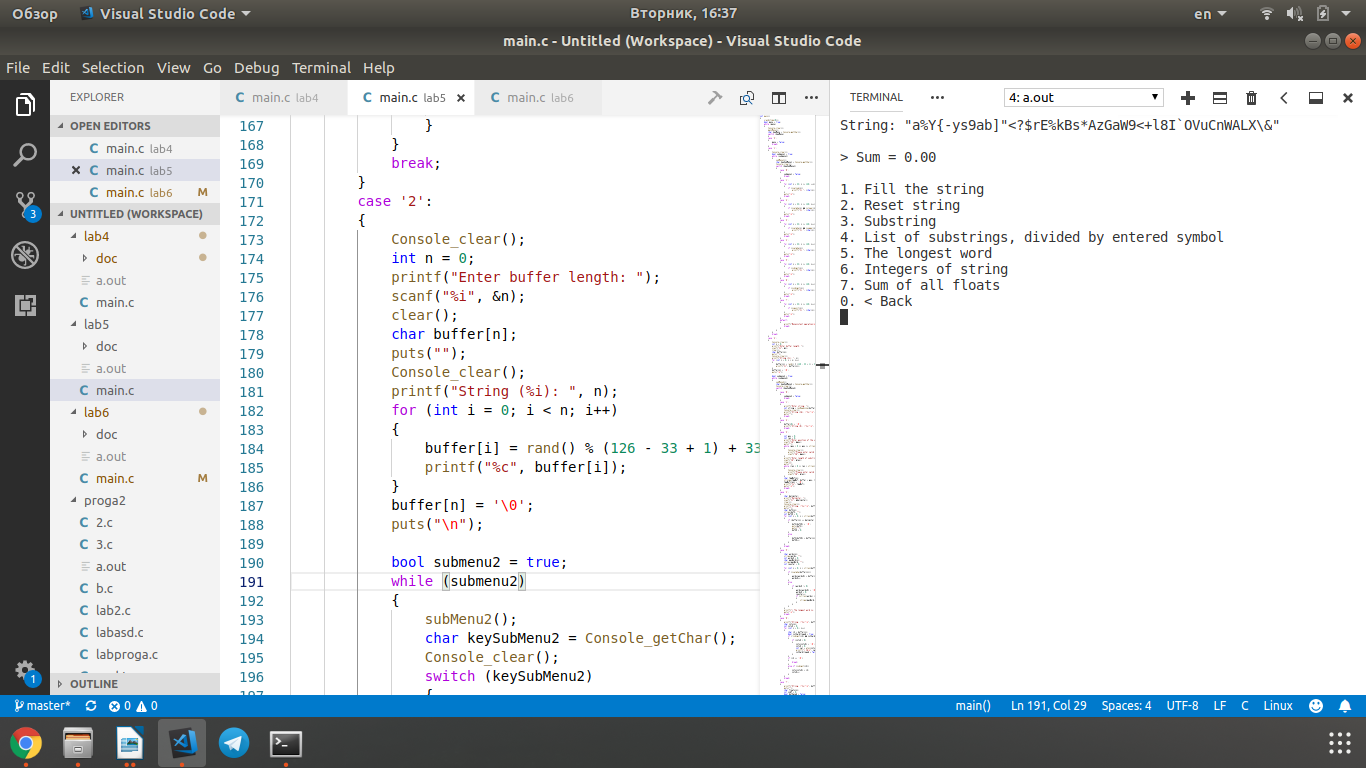
У другому підменю потрібно задати довжину буфера для зберігання строки:

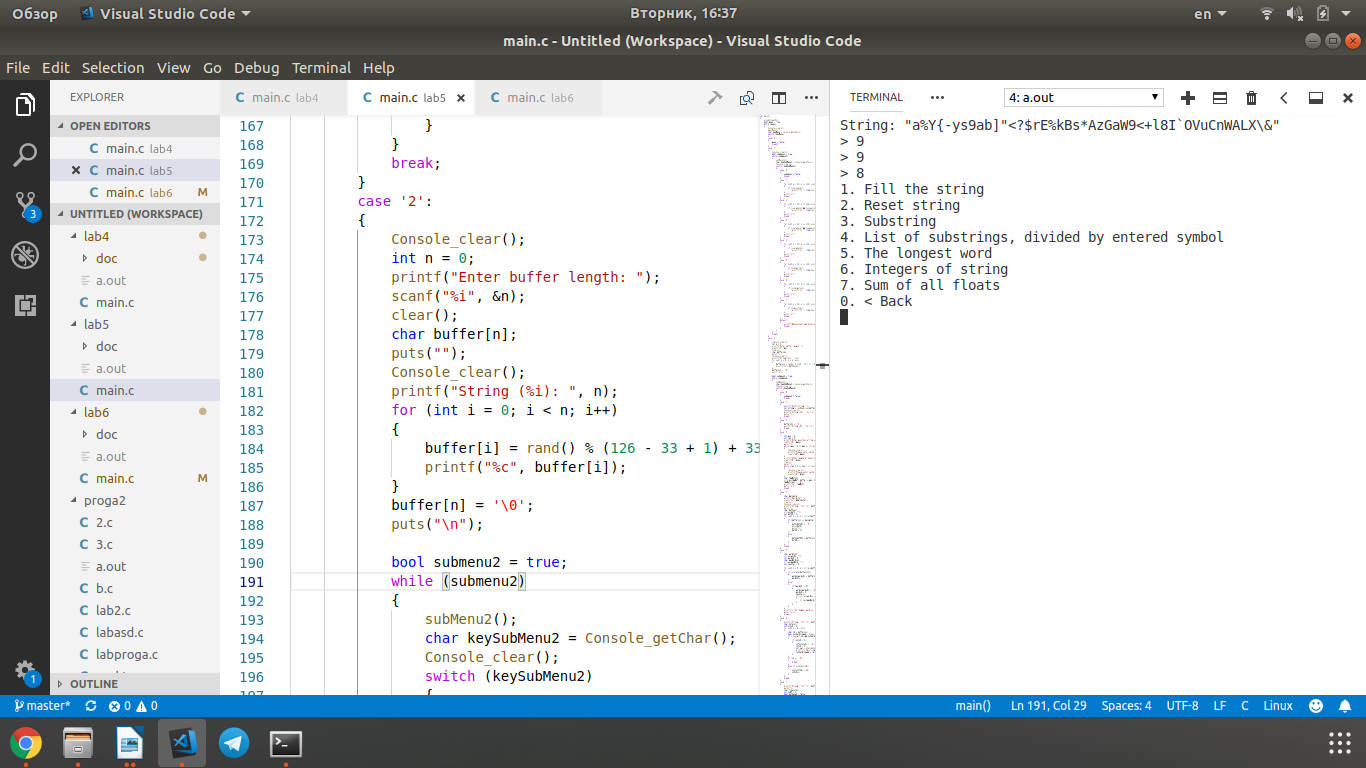


Доступні операції над строкою:



Декілька прикладів роботи програми:





**Висновок**

Протягом виконання цієї лабораторної роботи я ознайомилася із програмуванням на мові С у спеціалізованому редакторі коду.

Більше того, на практиці я навчилася працювати з масивами символів та рядками. Також я використовала стандартні функції для роботи з рядками для вирішення прикладних задач. Окрім цього, мною було засвоєно оформлення консольної програми для зручності роботи користувача.

Компіляція всього коду відбувалася за допомогою компілятора gcc.

Отже, мета лабораторної роботи досягнена, всі завдання виконані і їх розвʼязання наведені.