rr

**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”**

|  |  |  |
| --- | --- | --- |
| **Виконав**  **студент I курсу**  **групи КП-83**  **Мортіков Владислав Євгенович**  **варіант №16** |  | **Перевірив**  **“\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.**  **викладач**  **Гадиняк Руслан Анатолійович**  **(прізвище, ім’я, по батькові)** |

**Київ 2018**

**Мета роботи**

Навчитися працювати зі масивами символів та рядками. Уміти визначати класи символів та використовувати функції для роботи із символами.

Також, навчитись використовувати стандартні функції для роботи з рядками для вирішення прикладних задач та 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?/). Виводити із буфера рядок та його поточну довжину, наприклад:

String (5):   
>>>  
The !  
<<<

Доступні операції над рядком:

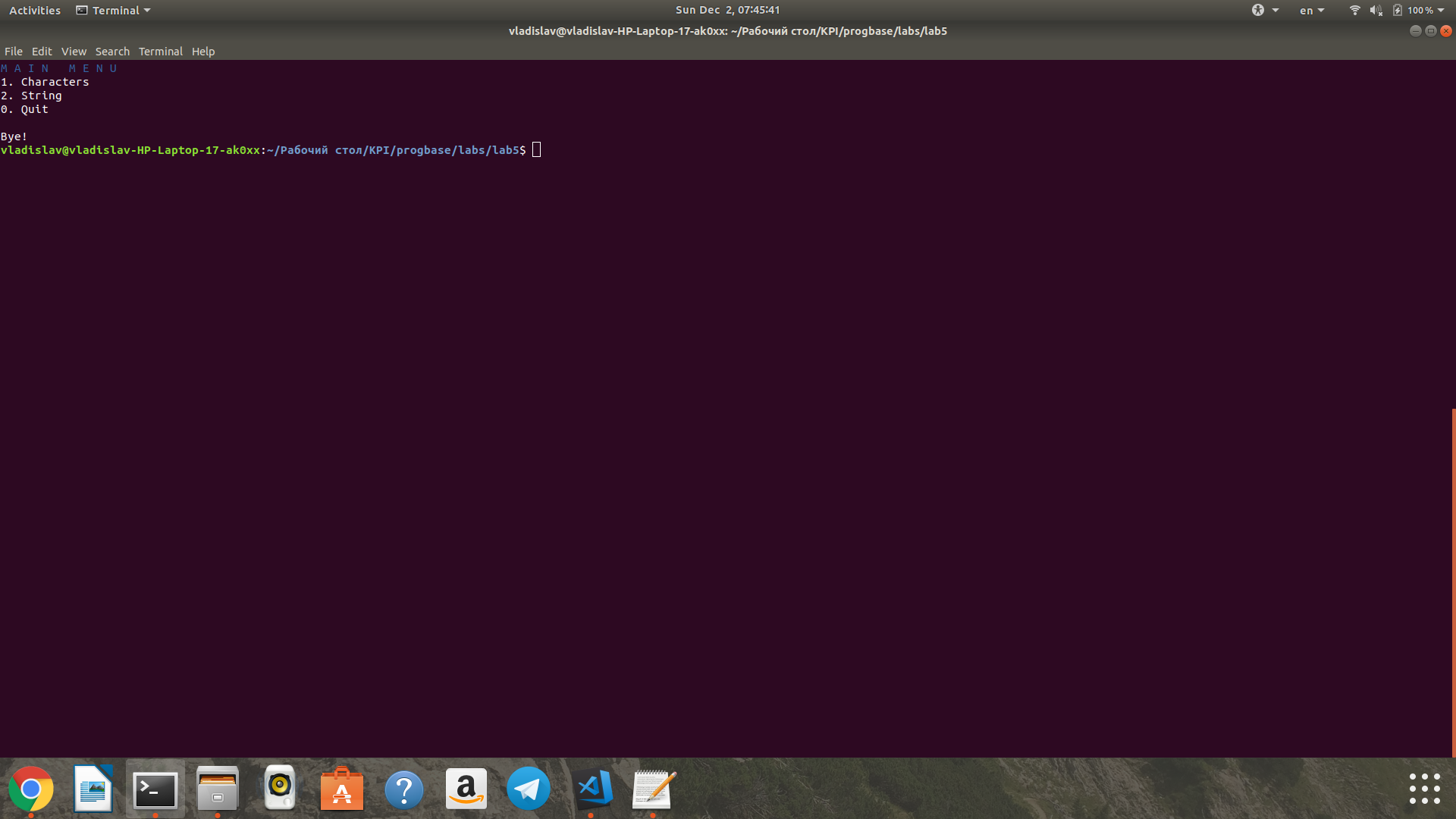
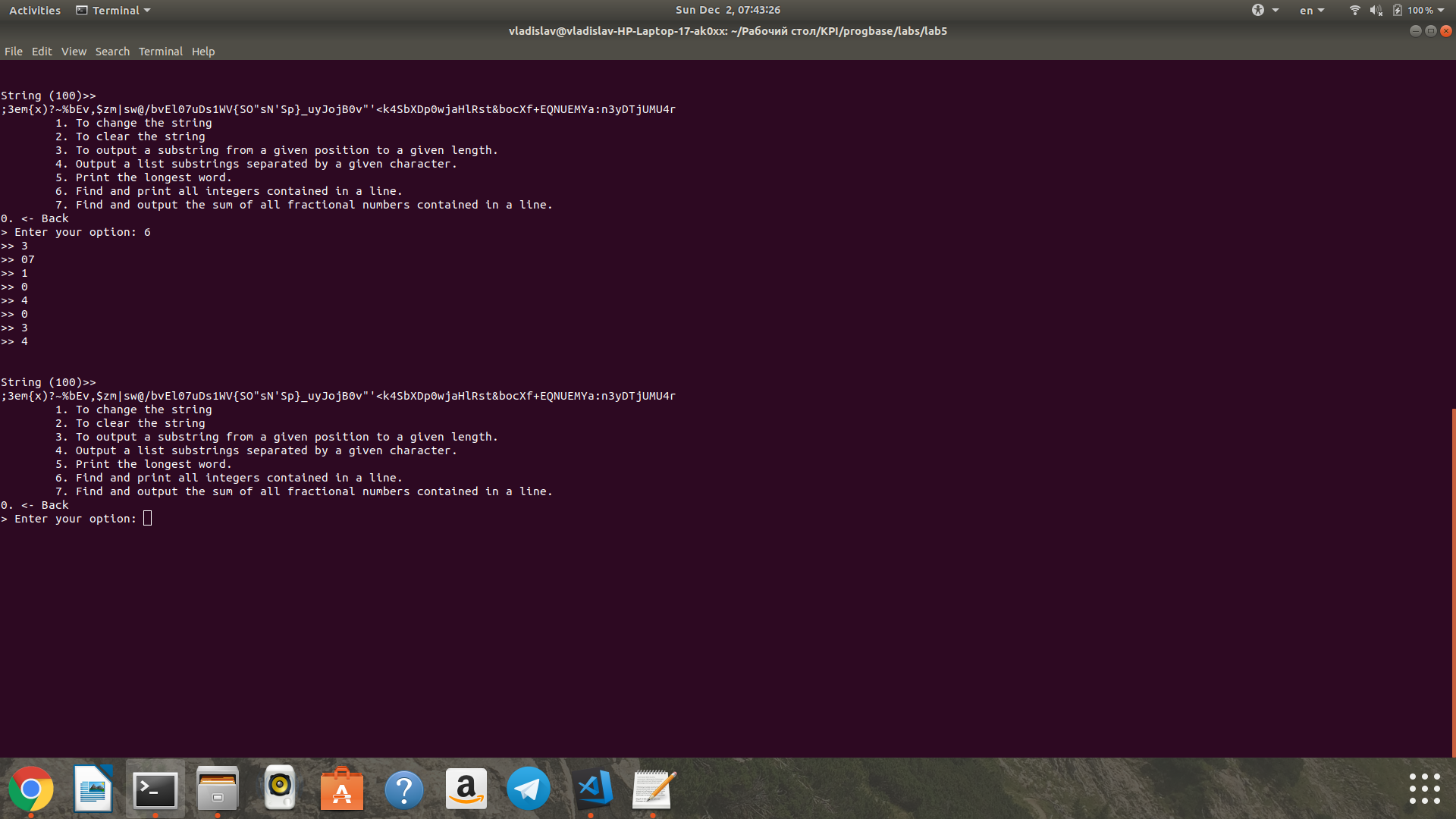
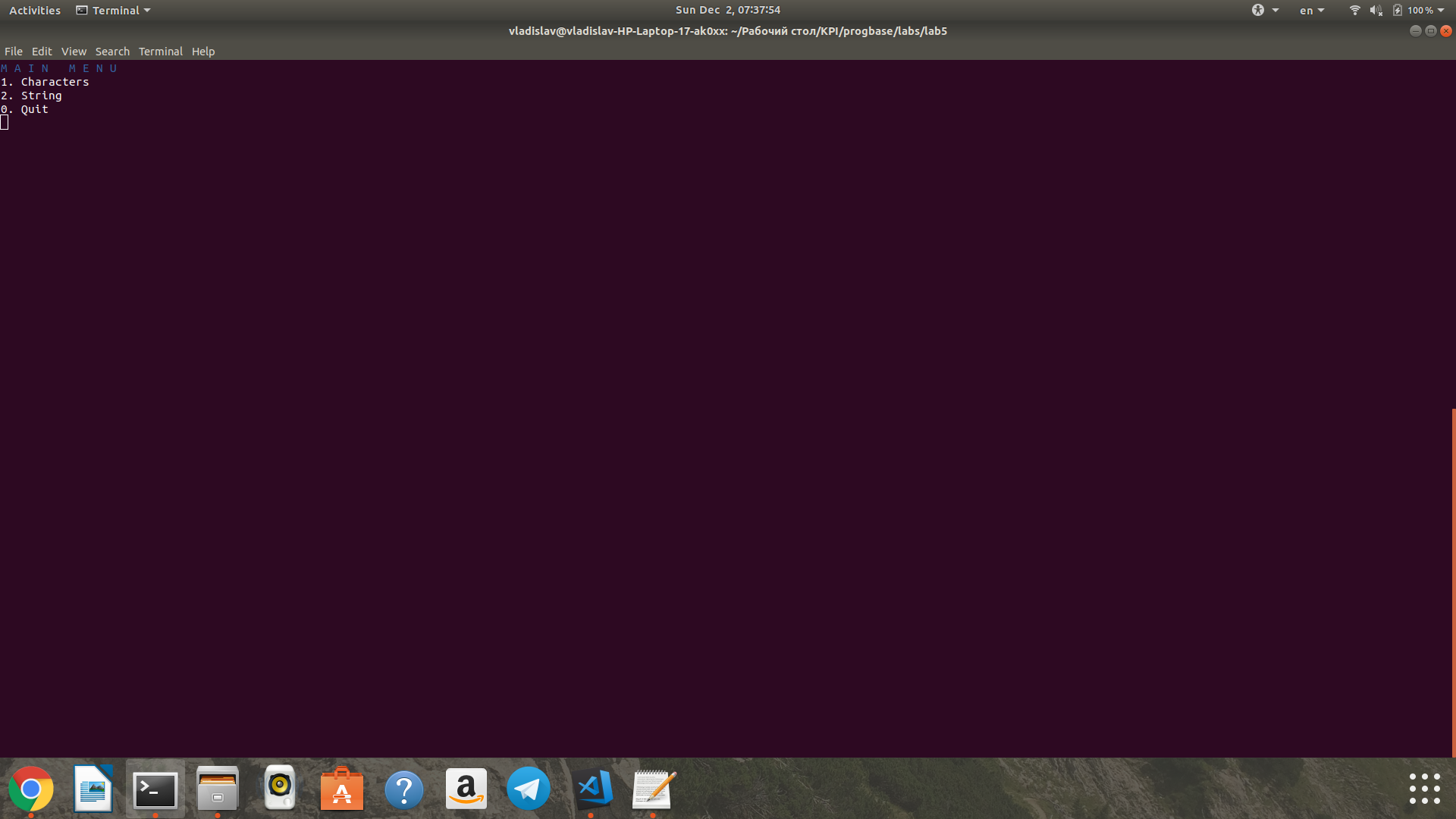
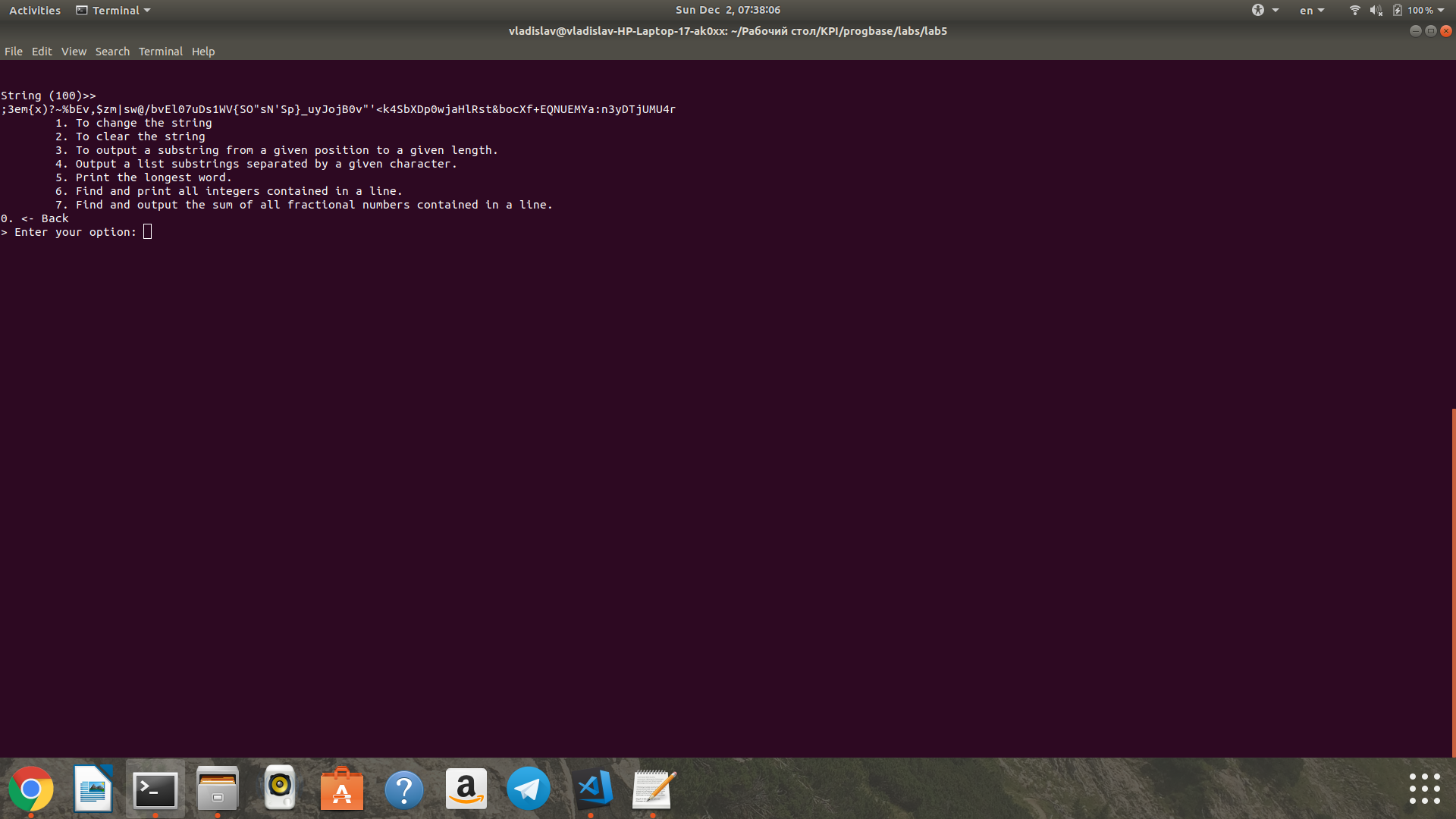
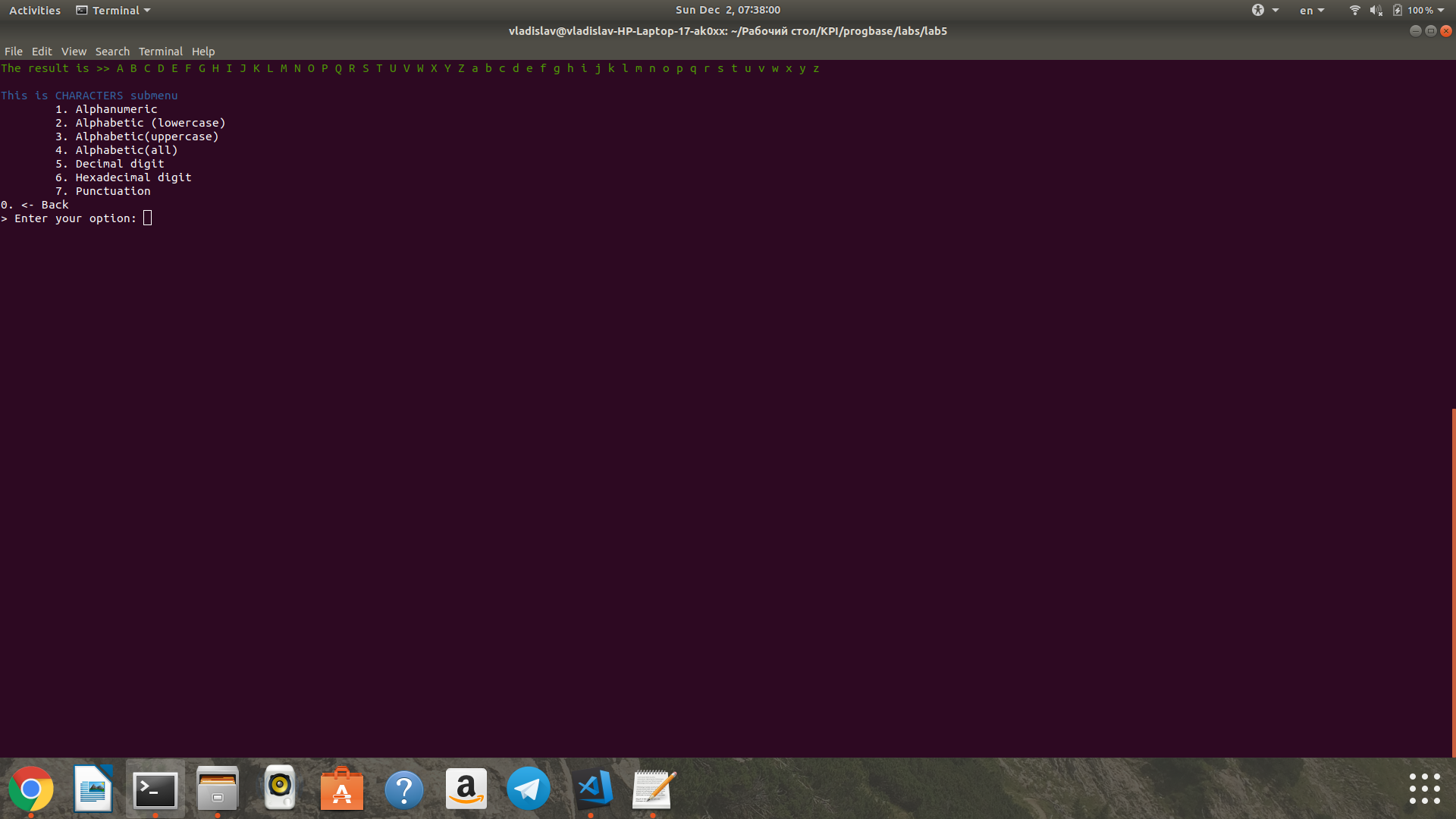
1. Замінити рядок у масиві на введене рядкове значення із консолі (рядок може містити введені пробіли, використовувати fgets(), UPD-4 див. Додатки).
2. Очистити рядок (UPD зробити його пустим, довжина рядка має стати 0).
3. Вивести підрядок із заданої позиції і заданої довжини.
4. Вивести список (UPD-5 непустих) підрядків, розділених заданим символом.
5. (UPD-5) Приклади:
   * рядок What I123have 10a do?, символ *пробіл*, підрядки: What, I123have, 10a, do?.
   * рядок What I123have 10a do?, символ a, підрядки: Wh, t I123h, ve 10, do?.
   * рядок Hi!!, символ !, підрядки: Hi.
   * рядок Hi, символ F, підрядки: Hi.
   * рядок *пустий*, символ *будь-який*, підрядки: *відсутні*.
6. Вивести найдовше слово.
7. Знайти та вивести всі цілі числа, що містяться у рядку.
8. Знайти та вивести суму всіх дробових чисел, що містяться у рядку.

Усі 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.

* Слово - непуста неперервна послідовність буквенних символів: Hi 123all'there, слова Hi, all, there.
* Цілі числа - неперервна послідовність десяткових цифр: g34.45ji456 -4, числа: 34, 45, 456, 4.
* Дробові числа - неперервна послідовність десяткових цифр, серед яких знаходиться одна точка (точка може бути початком або кінцем такої послідовності): g34.45ji456 -4.aa.3k23!12.23.34, числа: 34.45, 4.0, 0.3, 12.23, 0.34.

**Тексти коду програм**

|  |
| --- |
| **main.c** |
| #include <stdio.h>  #include <math.h>  #include <stdlib.h>  #include <progbase.h>  #include <time.h>  #include <stdbool.h>  #include <progbase/console.h>  #include <ctype.h>  #include <string.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 title1STmenu()  {  Console\_setCursorAttribute(FG\_GREEN);  printf("\nThe result is >> ");  }  int main()  {  srand(time(0));  bool mainMenu = true;  while (mainMenu)  {  Console\_clear();  Console\_setCursorAttribute(FG\_BLUE);  puts("M A I N M E N U");  Console\_reset();  puts("1. Characters");  puts("2. String");  puts("0. Quit ");  char mainMenuUserInput = Console\_getChar();  switch (mainMenuUserInput)  {  case '0':  {  mainMenu = false;  break;  }  case '1':  {  bool isSubMenu1Running = true;  while (isSubMenu1Running)  {  Console\_clear();  Console\_setCursorAttribute(FG\_BLUE);  puts("\n\nThis is CHARACTERS submenu");  Console\_reset();  puts("\t1. Alphanumeric");  puts("\t2. Alphabetic (lowercase)");  puts("\t3. Alphabetic(uppercase) ");  puts("\t4. Alphabetic(all)");  puts("\t5. Decimal digit");  puts("\t6. Hexadecimal digit");  puts("\t7. Punctuation");  puts("0. <- Back");  printf("> Enter your option: ");  char menu1Running = Console\_getChar();  printf("You entered: '%c'\n", menu1Running);  if (menu1Running > 7 && menu1Running < 0)  {  printf("\nPlease, choose numbers from '0' to '7'");  }  else  switch (menu1Running)  {  case '0':  {  isSubMenu1Running = false;  }  break;  case '1':  {  title1STmenu();  for (int c = 0; c <= 200; c++)  {  if (isalnum(c))  {  putchar(c);  putchar(32);  }  }  }  break;  case '2':  {  title1STmenu();  for (int c = 0; c <= 200; c++)  {  if (isalpha(c))  {  if (islower(c))  {  putchar(c);  putchar(32);  }  }  }  }  break;  case '3':  {  title1STmenu();  for (int c = 0; c <= 200; c++)  {  if (isalpha(c))  {  if (isupper(c))  {  putchar(c);  putchar(32);  }  }  }  }  break;  case '4':  {  title1STmenu();  for (int c = 0; c <= 200; c++)  {  if (isalpha(c))  {  putchar(c);  putchar(32);  }  }  }  break;  case '5':  {  title1STmenu();  for (int c = 0; c <= 200; c++)  {  if (isdigit(c))  {  putchar(c);  putchar(32);  }  }  }  break;  case '6':  {  title1STmenu();  for (int c = 0; c <= 200; c++)  {  if (isxdigit(c))  {  putchar(c);  putchar(32);  }  }  }  break;  case '7':  {  title1STmenu();  for (int c = 0; c <= 200; c++)  {  if (ispunct(c))  {  putchar(c);  putchar(32);  }  }  }  break;  }  }  case '2':  {  Console\_clear();  int stringLen = 0;  printf("Please, input a size of the string >> \n");  scanf("%i", &stringLen);  Console\_clear();  char string[stringLen];  int key = 0;  for (int c = 0; c < stringLen; c++)  {  string[c] = rand() % (126 - 32 + 1) + 32;  }  string[stringLen - 1] = '\0';  do  {  printf("\n\nString (%i)>>\n", stringLen);  puts(string);  puts("\t1. To change the string");  puts("\t2. To clear the string");  puts("\t3. To output a substring from a given position to a given length.");  puts("\t4. Output a list substrings separated by a given character.");  puts("\t5. Print the longest word.");  puts("\t6. Find and print all integers contained in a line.");  puts("\t7. Find and output the sum of all fractional numbers contained in a line.");  puts("0. <- Back");  printf("> Enter your option: ");  scanf("%i", &key);  for (char ch; (ch = getchar()) != '\n';)  {  }  switch (key)  {  case 1:  {  Console\_clear();  printf("Enter string: ");  const int bufLen = stringLen;  char buf[bufLen];  int strLen = 0;  while (!strLen)  strLen = conReadLine(buf, bufLen);  stringLen = strLen;  for (int c = 0; c <= stringLen; c++)  {  string[c] = buf[c];  if (c == strLen)  {  string[c] = '\0';  }  }  printf("Entered (%d): \"%s\"", strLen, buf);  puts("\n---");  }  break;  case 2:  {  Console\_clear();  string[0] = '\0';  }  break;  case 3:  {  int positionMin = 0;  int positionMax = 0;  puts("Input min:");  scanf("%i", &positionMin);  puts("Input max:");  scanf("%i", &positionMax);  Console\_clear();  if (positionMin - 1 >= 0 && positionMax < stringLen)  {  printf(">>Substring:\n");  for (int c = positionMin; c <= positionMax; c++)  {  putchar(string[c]);  }  puts(" ");  }  else  {  break;  }  }  break;  case 4:  {  Console\_clear();  char separator = 0;  scanf("%c", &separator);  int tmpI = 0;  char tmpStr[stringLen];  for (int i = 0;; i++)  {  char ch = string[i];  if (ch == separator || ch == '\0')  {  if (tmpI != 0)  {  tmpStr[tmpI] = '\0';  printf(">> %s", tmpStr);  puts("");  tmpI = 0;  }  if (ch == '\0')  {  break;  }  }  else  {  tmpStr[tmpI] = ch;  tmpI += 1;  }  }  }  break;  case 5:  {  Console\_clear();  int tmpI = 0;  char maxlength[stringLen];  char tmpStr[stringLen];  maxlength[0] = '\0';  for (int i = 0;; i++)  {  char ch = string[i];  if (ch == '\0' || (isalpha(ch) == 0))  {  if (tmpI != 0)  {  tmpStr[tmpI] = '\0';  if (strlen(tmpStr) > strlen(maxlength))  {  strcpy(maxlength, tmpStr);  }  tmpI = 0;  }  if (ch == '\0')  {  break;  }  }  else if (isalpha(ch))  {  tmpStr[tmpI] = ch;  tmpI += 1;  }  }  puts(maxlength);  puts("\n");  }  break;  case 6:  {  int tmpI = 0;  char tmpStr[stringLen];  for (int i = 0;; i++)  {  char ch = string[i];  if (ch == '\0' || (isdigit(ch) == 0))  {  if (tmpI != 0)  {  tmpStr[tmpI] = '\0';  printf(">> %s\n", tmpStr);  tmpI = 0;  }  if (ch == '\0')  {  break;  }  }  else if (isdigit(ch))  {  tmpStr[tmpI] = ch;  tmpI += 1;  }  }  }  break;  case 7:  {  Console\_size();  int tmpI = 0;  char tmpStr[stringLen];  bool dotsaved = false;  float res = 0;  for (int i = 0;; i++)  {  char ch = string[i];  if (ch == '\0' || ((isdigit(ch) == 0) && dotsaved))  {  if (tmpI != 0)  {  tmpStr[tmpI] = '\0';  printf(">> %s\n", tmpStr);  res += atof(tmpStr);  tmpI = 0;  dotsaved = !dotsaved;  }  if (ch == '\0')  {  break;  }  }  else if ((isdigit(ch) && dotsaved) || (!dotsaved && ch == '.') || (!dotsaved && isdigit(ch)))  {  tmpStr[tmpI] = ch;  tmpI += 1;  if (ch == '.')  {  dotsaved = true;  }  }  }  printf(" the summ is %f\n", res);  }  break;  default:  {  Console\_clear();  }  break;  }  } while (key != 8);  }  }  }  }  puts("\nBye!");  return 0;  } |

**￼Приклади результатів**

**Висновки**

Виконавши дану лабораторну роботу ми навчилися працювати із масивами символів та рядками, визначати класи символів та використовувати функції для роботи із символами, змогли використовувати стандартні функції для роботи з рядками для вирішення прикладних задач, а також навчилися оформлювати консольну програму для зручності роботи користувача.

Вимушений констатувати, що в моїй реалізації даної лабораторної роботи є багато недоліків. Так, не проводится перевірка на валідність введених даних, правильно не виконується пункт 2.7.

Із часом, дана проблема буде виправлена, я на це сподіваюсь.

Компіляція всього коду відбувалася за допомогою компілятора gcc.