

Laboratorium 5

Przekazywanie parametrów do funkcji przez wartość, przez wskaźnik (adres), przez referencję (zmienną) oraz operacje na łańcuchach znaków

Program przykładowy nr 1

Proszę przeanalizować program przykładowy *row_kwa.cpp*, który ilustruje sposób podziału programu na kilka funkcji. Program oblicza pierwiastki równania kwadratowego.

Funkcje *CzytajDane* oraz *WypiszPierwiastki* zapewniają komunikację z użytkownikiem programu tzn. wczytują współczynniki równania, wypisują pierwiastki i inne komunikaty. Sposób implementacji tych funkcji zależy od wykorzystywanej biblioteki wejścia-wyjścia. W tym programie została wykorzystana biblioteka *stdio.h*, a do wypisywania i wczytywania danych zostały użyte funkcje *printf* oraz *scanf*.

Funkcje *Delta* oraz *ObliczPierwiastki* wykonują wszystkie obliczenia. Te funkcje nie wypisują żadnych komunikatów na ekranie, a więc są niezależne od wykorzystywanej biblioteki wejścia-wyjścia.

Proszę zwrócić szczególną uwagę na dobór sposobu przekazywania parametrów w poszczególnych funkcjach.

Zadanie 1

Proszę napisać program który rozwiązuje układ dwóch równań liniowych :

$$\begin{cases} a1 * x + b1 * y = c1 \\ a2 * x + b2 * y = c2 \end{cases}$$

Proszę zastosować metodę wyznaczników (Wzory Cramera):

$$W = \begin{vmatrix} a1 & b1 \\ a2 & b2 \end{vmatrix} = a1 * b2 - a2 * b1$$

$$Wx = \begin{vmatrix} c1 & b1 \\ c2 & b2 \end{vmatrix} = c1 * b2 - c2 * b1$$

$$Wy = \begin{vmatrix} a1 & c1 \\ a2 & c2 \end{vmatrix} = a1 * c2 - a2 * c1$$

Możliwe są trzy przypadki:

- Jeżeli $W \neq 0$ to układ posiada dokładnie 1 jedno rozwiązanie:

$$x = \frac{Wx}{W} \quad y = \frac{Wy}{W}$$

- Jeżeli $W = 0$ $Wx = 0$ $Wy = 0$ to układ posiada nieskończenie wiele rozwiązań.
- Jeżeli $W = 0$ $Wx \neq 0$ $Wy \neq 0$ to układ jest sprzeczny tzn. nie posiada żadnego rozwiązania.

Proszę podzielić program na funkcje realizujące komunikację z użytkownikiem oraz funkcje wykonujące obliczenia. Proponuję następujące funkcje:

```
void CzytajRownanie(float &a, float &b, float &c);
```

- funkcja wczytuje współczynniki jednego równania. $a*x+b*y=c$

```
void WypiszRozwiązanie( int N, float x, float y);
```

- funkcja wypisuje rozwiązanie układu równań lub inny komunikat

```
float ObliczWyznacznik(float p1, float p2, float p3, float p4);
```

-funkcja oblicza wyznacznik $\begin{vmatrix} p1 & p2 \\ p3 & p4 \end{vmatrix} = p1*p4 - p3*p2$

```
int ObliczRozwiązanie(float a1, float b1, float c1,
                     float a2, float b2, float c2,
                     float &x, float &y);
```

- funkcja oblicza rozwiązanie układu równań i zwraca:

2 – nieskończenie wiele rozwiązań

1 – jest jedno rozwiązanie

0 – brak rozwiązań (układ sprzeczny)

Funkcja *main* może mieć wówczas następującą postać:

```
Int main(int argc, char*argv[ ] )
{ float A1, B1, C1; // pierwsze równanie,
  float A2, B2, C2; // drugie równanie,
  float X, Y;       // rozwiązanie,
  int N;            // liczba rozwiązań.

  CzytajRownanie(A1, B1, C1);
  CzytajRownanie(A2, B2, C2);
  N = ObliczRozwiązanie(A1, B1, C1, A2, B2, C2, X, Y);
  WypiszRozwiązanie( N, X, Y);
  getch();
  return 0;
}
```

Zadanie 1a (dla ambitnych)

Proszę zmodyfikować program z poprzedniego zadania tak, by parametry przekazywane do funkcji przez referencję zastąpić parametrami przekazywanymi przez wskaźnik.

Uwaga: Zmodyfikowany program proszę umieścić nowym pliku źródłowym.

Zadanie 2

Proszę zapoznać się z prototypami, opisami i przykładami działania następujących funkcji operujących na łańcuchach znaków:

```
int strlen ( const char * str );
```

```
int strcmp ( const char * str1, const char * str2 );
```

```

char * strcpy ( char * destination, const char * source );
char * strncpy ( char * destination, const char * source, size_t num );

char * strcat ( char * destination, const char * source );
char * strncat ( char * destination, char * source, size_t num );

char * strstr ( char * str1, const char * str2 );

char * strchr ( const char * str, int character );
char * strrchr ( char * str, int character );

int strspn ( const char * str1, const char * str2 );
int strcspn ( const char * str1, const char * str2 );
char * strpbrk ( char * str1, const char * str2 );
char * strtok ( char * str, const char * delimiters );

```

Opisy i przykłady działania można znaleźć np. na stronie

<http://www.cplusplus.com/reference/cstring/>

Proszę napisać własne wersje co najmniej 4 funkcji wybranych spośród funkcji podanych powyżej, które będą wykonywały takie same operacje co ich standardowe odpowiedniki z biblioteki `string.h`, np.:

```

int moje_strlen ( const char * str );
int moje_strcmp ( const char * str1, const char * str2 );
char * moje_strcpy ( char * destination, char * source );
char * moje_strchr ( const char * str, int character );

```

W funkcji `main` proszę napisać przykłady użycia napisanych funkcji, które pokazują, że napisane funkcje działają tak samo jak ich standardowe odpowiedniki z biblioteki `string.h`. Przykłady można zrobić podobnie jak przykłady zamieszczone na stronie

<http://www.cplusplus.com/reference/cstring/>

Zadanie 3

Proszę uzupełnić program z zadanie 2 i napisać własne wersje funkcji `strupr` oraz `strlwr`

```

char* moje_strupr(char* );
char* moje_strlwr(char* );

```

które w podanym łańcuchu znaków zamieniają wszystkie litery na duże (funkcja `strupr`) lub małe (funkcja `strlwr`). Znaki nie będące literami nie powinny być modyfikowane.

W funkcji `main` proszę napisać przykłady użycia napisanych funkcji,

Zadanie 2a,3a (dla ambitnych)

Osoby ambitne mogą napisać wszystkie funkcje z zadanie 2 i 3 w dwóch wariantach. W pierwszym wariacie odwołania do poszczególnych znaków w przetwarzanych łańcuchach proszę robić za pomocą wskaźników. W drugim wariacie odwołania do poszczególnych znaków proszę robić za pomocą indeksów, traktując przetwarzane łańcuchy jako tablice znaków, których adres pierwszego elementu jest podany jako parametr funkcji.