

Imię i nazwisko	Kierunek	Rok studiów i grupa
Marek Kubicki	Informatyka techniczna ITE	1 rok grupa 5
Data zajęć	Numer i temat sprawozdania	
03.11.2022	Lab 5. – Konstrukcje sterujące	

Cel:

- Nauka posługiwania się konstrukcjami typu switch, poznanie sposobu na porównywanie liczb zmiennoprzecinkowych.

Przebieg zajęć:

Utworzyłem katalog roboczy lab_5 a w nim podkatalogi switch i rownanie_kwadratowe.

```
m@m-VirtualBox:~/Lab_5/switch$ ls
if.c ifpierw.c simple_switch.c skr1 skr2 skr3
m@m-VirtualBox:~/Lab_5/switch$ pwd
/home/m/Lab_5/switch
m@m-VirtualBox:~/Lab_5/switch$ S
```

```
m@m-VirtualBox:~/Lab_5/równanie_kwadratowe$ ls
kw równanie_kwadratowe.c skr3.asc
m@m-VirtualBox:~/Lab_5/równanie_kwadratowe$ pwd
/home/m/Lab_5/równanie_kwadratowe
m@m-VirtualBox:~/Lab_5/równanie_kwadratowe$
```

Następnie skopiowałem do nich odpowiednio programy simple_switch.c oraz równanie_kwadratowe.c i kompliluj.sh.

Zmieniłem konstrukcje programu switch.c w taki sposób aby każdy klawisz wywoływał tylko jeden zestaw operacji (odpowiedź dla 1-5 zakończenie programu dla 6 i odpowiedź na resztę).

```

switch (c) {
    case '0':
        printf("Wprowadzono: 0\n");
        break;
    case '1':
        printf("Wprowadzono: 1\n");
        break;
    case '2':
        printf("Wprowadzono: 2\n");
        break;
    case '3':
        printf("Wprowadzono: 3\n");
        break;
    case '4':
        printf("Wprowadzono: 4\n");
        break;
    case '5':
        printf("Wprowadzono: 5\n");
        break;
    case '6':
        exit(0);
    default:
        printf("Wprowadzono: znak spoza zakresu 0-6\n");
        break;
}

Wprowadź cyfrę od 0 do 5:
2
Wprowadzono: 2

Wprowadź cyfrę od 0 do 5:
3
Wprowadzono: 3

Wprowadź cyfrę od 0 do 5:
4
Wprowadzono: 4

Wprowadź cyfrę od 0 do 5:
1
Wprowadzono: 1

Wprowadź cyfrę od 0 do 5:
6

```

Następnie utworzyłem plik if.c którego działanie było identyczne do działania pliku switch.c poza tym że opierał je na konstrukcji if ... elseif ... else.

```

if(c==0)
{
    printf("Wprowadzono: 0\n");
}
else if(c==1)
{
    printf("Wprowadzono: 1\n");
}
else if(c==2)
{
    printf("Wprowadzono: 2\n");
}
else if(c==3)
{
    printf("Wprowadzono: 3\n");
}
else if(c==4)
{
    printf("Wprowadzono: 4\n");
}
else if(c==5)
{
    printf("Wprowadzono: 5\n");
}
else if (c==6)
{
    break;
}
else
{
    printf("Wprowadzono: znak spoza zakresu 0-6\n");
}

./s

```

Następnie zmodyfikowałem program rownanie_kwadratowe.c w taki sposób aby uniknąć porównywania wartości zmiennoprzecinkowych a zamiast tego sprawdzać czy błąd wynikający z ich niedokładności jest mniejszy od tolerancji

```
#define TOLERANCJA 1.e-14
```

Dodałem także fragment kodu sprawdzający to czy wynik który został obliczony mieści się w zakresie tolerancji.

```
    if(fabs((-b-temp)/(2*a)*(-b+temp)/(2*a) - (c/a))<TOLERANCJA)
{
    printf("Pierwiastki sa dokladne\n");
}
else
{
    printf("Pierwiastki nie sa dokladne\n");
}
```

Przykładowe porównanie:

```
(fabs(delta) < TOLERANCJA)
```

```
Program rozwiązywania równania kwadratowego ax^2 + bx + c = 0
Podaj parametr a: 1
Podaj parametr b: 2
Podaj parametr c: 1
Jeden pierwiastek rzeczywisty: x = -1.000000000000e+00
Pierwiastki sa dokladne
```

```
Podaj parametr a: 10
Podaj parametr b: 50
Podaj parametr c: 4
Dwa pierwiastki rzeczywiste: x1 = -4.918677324490e+00, x2 = -8.132267551044e-02
Sprawdzenie poprawnosci Viete: x1*x2=c/a
Pierwiastki nie sa dokladne
```

Dodałem także możliwość obliczania wyników będących liczbami zespolonymi

```
else if(delta<0)
{
    SCALAR temp = sqrt(-delta);
    printf("Dwa pierwiastki zespolone: x1 = %.12le + i%.12le, x2 = %.12le
    - i%.12le \n",(-b)/(2*a), (temp/(2*a)), (-b)/(2*a), (temp/(2*a)));
}
```

```
Podaj parametr a: 10
Podaj parametr b: -20
Podaj parametr c: 40
Dwa pierwiastki zespolone: x1 = 1.00000000000e+00 + i1.732050807569e+00, x2 =
1.00000000000e+00 - i1.732050807569e+00
Pierwiastki nie sa dokladne
```

Napisałem też zmodyfikowaną wersję wcześniejszego programu opierającego się na operacjach typu: if ... elseif ... else, który zachowywał się jak pierwotna wersja programu simple_switch.c

Wprowadź cyfrę od 0 do 5: 1 Wprowadzono: 1 lub 2	if(c==0) { printf("Wprowadzono: 0\n"); } else if(c==1 c==2) { printf("Wprowadzono: 1 lub 2\n"); } else if(c==3) { printf("Wprowadzono: 3\n"); } else if(c==4 c==5) { printf("Wprowadzono: 4 lub 5\n"); } else [printf("Wprowadzono: znak spoza zakresu 0-5\n");]
Wprowadź cyfrę od 0 do 5: 2 Wprowadzono: 1 lub 2	
Wprowadź cyfrę od 0 do 5: 3 Wprowadzono: 3	
Wprowadź cyfrę od 0 do 5: 4 Wprowadzono: 4 lub 5	
Wprowadź cyfrę od 0 do 5: 5 Wprowadzono: 4 lub 5	
Wprowadź cyfrę od 0 do 5: 6 Wprowadzono: znak spoza zakresu 0-5	
Wprowadź cyfrę od 0 do 5:	

Wnioski

Zmienne w języku C muszą być wykorzystywane z zachowaniem ostrożności. Gdy nie uważa się na to jak dokładnie działa program to np.: float posiadający niedokładność (np. swoje ostatnie liczby) może zmienić działanie programu. Konstrukcje switch, jeżeli dobrze przemyślane i skonstruowane, ułatwiają pisanie kodu i rozgałęziających się ścieżek programu.