

Imię i nazwisko	Kierunek	Rok studiów i grupa
Marek Kubicki	Informatyka techniczna ITE	1 rok grupa 5
Data zajęć	Numer i temat sprawozdania	
27.10.2022	Lab 4. - Zmienne i operatory	

Cel:

- Opanowanie tworzenia prostych programów w C zapisujących wartości zmiennych różnych typów oraz wykonujących podstawowe operacje na zmiennych.

Przebieg zajęć:

Utworzyłem katalog roboczy lab_4 i pobrałem plik zmienne.c. Po od komentowaniu wykonałem kolejne polecenia.

Pierwszym poleceniu zadeklarowałem kilka zmiennych

```
int n1 = -1;
float f1 = 0.5;
double d1 = -0.125; //long float
long double ld1 = 0.0625;
long int ln1 = -1000;
char c1 = 'c';
char string[3] = {'s', 'a', 'm'};
unsigned int ui1 = 20000;
```

Oraz wypisałem je do terminala poprawnie:

```
printf ("\\nint = %d \\nfloat = %f \\ndouble = %lf \\nlong double = %llf \\nlong int = %ld
\\nunsigned int = %u \\nchar = %c \\nstring(char[]) = %s", n1, f1, d1, ld1, ln1, ui1, c1, string);
```

A potem zmieniłem typy wypisywania tych samych zmiennych:

```
printf ("\\nint = %f \\nfloat = %ld \\ndouble = %llf \\nlong double = %c \\nlong int = %d
\\nunsigned int = %d \\nchar = %u \\nstring(char[]) = %d\\n", n1, f1, d1, ld1, ln1, ui1, c1, string);
```

```
mc [markubi2@WL31112:~/lab_4]
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
int = -1
float = 0.500000
double = -0.125000
long double = 0.062500
long int = -1000
unsigned int = 20000
char = c
string(char[]) = same

int = 0.500000
float = 4294967295
double = 0.062500
long double = 0.062500
long int = 20000
unsigned int = 99
char = 2669308384
string(char[]) = -1625658880
```

```
Program dotarł do końca  
sh-4.2$ pwd  
/home/METAL/markubi2/lab_4  
sh-4.2$ ls  
skrypt4 _zmienne.c
```

Kolejne polecenie polegało na zaznajomieniu się z działaniem kolejności wykonywania działań. W tym celu zadeklarowałem kilka zmiennych i sprawdziłem co stanie się kiedy w różny sposób ujmę działania na nich wykonane w nawiasy.

```
int a2 = 3;  
int b2 = 7;  
int c2 = 4;  
int wn12= a2+b2*c2;  
int wn22= (a2+b2)*c2;  
printf("\n3+7*4 = %d \n(3+7)*4 = %d \n", wn12, wn22);
```

```
|3+7*4 = 31  
|(3+7)*4 = 40
```

W następnym zadaniu trzeba było sprawdzić inkrementacja zapisana przed zmienną z tą zapisaną po niej po niej. W tym celu wykorzystałem oba sposoby inkrementacji w prostych działaniach i sprawdziłem wyniki.

```
int c3 = 3;  
int a3;  
a3 = c3++;  
c3--;  
int b3;  
b3 = ++c3;  
c3--;  
int wn13;  
wn13= (++a3)+b3;  
a3--;  
int wn23;  
wn23= a3+(b3++);  
b3--;  
printf("\n3++ = %d \n++3 = %d \n(++3)+4 = %d \n3+(4++) = %d \n",a3, b3 , wn13, wn23);
```

```
3++ = 3  
++3 = 4  
(++3)+4 = 8  
3+(4++) = 7
```

W ostatnie zadanie dotyczące prostych zmiennych polegało na zaznajomieniu się z działaniem operacji dzielenia oraz modulo na liczbach typu int. W tym celu wykonałem kilka prostych operacji na zmiennych int.

```
int a4 = 40;
int b4 = 3;
int wn14 = 40 / 3;
int wn24 = 40 % 3;
int s4 = wn14*3+wn24;
printf("\n40/3 = %d \n40%3 = %d \n(40/3)*3+(40%3) = %d\n", wn14, wn24, s4);
```

```
40/3 = 13
40%3 = 1
(40/3)*3+(40%3) = 40
```

Zadanie dotyczące floatów polegało na zaobserwowaniu przybranej przez nich wartości jeżeli przyporządkuje im się liczbę taką jak np. 1/5 (nie wielokrotność 2). W tym celu przypisałem do floata taką właśnie wartość i wyświetliłem ją dwukrotnie. Najpierw z standardową dokładnością a potem ze zwiększoną (żeby zobaczyć efekt przypisania 1/5 do zmiennej float)

```
float f16 = 5;
float f26 = 1;
float f36 = f26/f16;
printf("\nfloat 1/5 = %f \n %20.15f", f36, f36);
```

```
float 1/5 = 0.200000
0.200000002980232
```

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.