


SPRAWOZDANIE NR 1

Nazwa ćwiczenia	Wstęp do języka C		 POLITECHNIKA BYDGOSKA Wydział Telekomunikacji, Informatyki i Elektrotechniki
Przedmiot	Podstawy programowania - laboratorium		
Student grupa	Marcin Ogórkiewicz, grupa 7		
Data ćwiczeń	12.10.2022	24.10.2022	Data oddania sprawozdania

```
1 #include <stdio.h>
2
3 void zad1() {
4     printf("Zadanie 1\n");
5     int n = 76;
6     printf("Ustalona liczba(76), w formacie int, wyświetlona w formacie float z jednym i trzema miejscami po przecinku oraz w formacie napisowym:\n%.1f\n%.3f\n%c\n", (float)n, (float)n, (char)n);
7 }
8
9 void zad2() {
10     int a;
11     printf("Zadanie 2\nPodaj kod ASCII aby otrzymać znak\n");
12     scanf(" %d", &a);
13     if(a < 0 || a > 255) {
14         do {
15             printf("Kody ASCII przyjmują jedynie wartości 0-255. Podaj odpowiednią wartość\n");
16             scanf(" %d", &a);
17         } while(a < 0 || a > 255);
18     }
19     char ascii = (char)a;
20     printf("Twoj znak to: %c\n", ascii);
21 }
22
23 void zad3() {
24     float m, h;
25     printf("Zadanie 3: obliczanie BMI\nPodaj swoją wagę[kg] oraz wzrost[cm] według wzoru: masa, wzrost\n");
26     scanf(" %f, %f", &m, &h);
27     if(m <= 0 || h <= 0) {
28         do {
29             printf("Nie można ważyć 0 bądź mniej kilo, ani mieć 0 lub mniej centymetrow wzrostu. Podaj poprawną wagę[kg] oraz wzrost[cm]\n");
30             scanf(" %f, %f", &m, &h);
31         } while(m <= 0 || h <= 0);
32     }
33 }
```

```
1 #include <stdio.h>
2
3 void zad1() {
4     printf("Zadanie 1\n");
5     int n = 76;
6     printf("Ustalona liczba(76), w formacie int, wyświetlona w formacie float z jednym i trzema miejscami po przecinku oraz w formacie napisowym:\n%.1f\n%.3f\n%c\n", (float)n, (float)n, (char)n);
7 }
8
9 void zad2() {
10     int a;
11     printf("Zadanie 2\nPodaj kod ASCII aby otrzymać znak\n");
12     scanf(" %d", &a);
13     if(a < 0 || a > 255) {
14         do {
15             printf("Kody ASCII przyjmują jedynie wartości 0-255. Podaj odpowiednią wartość\n");
16             scanf(" %d", &a);
17         } while(a < 0 || a > 255);
18     }
19     char ascii = (char)a;
20     printf("Twoj znak to: %c\n", ascii);
21 }
22
23 void zad3() {
24     float m, h;
25     printf("Zadanie 3: obliczanie BMI\nPodaj swoją wagę[kg] oraz wzrost[cm] według wzoru: masa, wzrost\n");
26     scanf(" %f, %f", &m, &h);
27     if(m <= 0 || h <= 0) {
28         do {
29             printf("Nie można ważyć 0 bądź mniej kilo, ani mieć 0 lub mniej centymetrow wzrostu. Podaj poprawną wagę[kg] oraz wzrost[cm]\n");
30             scanf(" %f, %f", &m, &h);
31         } while(m <= 0 || h <= 0);
32     }
33 }
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help PodsProgSpr1 - main.c
Project: CMakelists.txt x main.c
31 }while(m <= 0 || h<=0);
32 }
33 h = h/100;
34 h*=h;
35 float bmi = m/h;
36 printf("Taoie BMI wynosi: %.2f\n", bmi);
37 }
38
39 void zad4(){
40     printf("Zadanie 4\nPodaj swoj wiek w latach, aby otrzymac jego wartosc w sekundach\n");
41     int lata;
42     scanf(" %d", &lata);
43     if(lata<=0){
44         do {
45             printf("Podaj NORMALNA ilosc lat(wieksza od zera)\n");
46             scanf(" %d", &lata);
47         } while(lata<=0);
48     }
49     float lata_pom = lata*3.156;
50     int notacja = 7;
51     if(lata_pom>=10){
52         do {
53             lata_pom/=10;
54             notacja++;
55         }while(lata_pom>=10);
56     }
57     printf("Tvoj wiek w sekundach, wyrazony w notacji wykladniczej, to:\n%f*10^%d\n", lata_pom, notacja);
58 }
59
60 int main() {
61     zad1();
62 }
63
64 Version Control Run Python Packages TODO Messages CMake Problems Terminal Services
Build finished in 317 ms (3 minutes ago) clang-tidy 3.13 LF UTF-8 4 spaces C:PodsProgSpr1 | Debug
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help PodsProgSpr1 - main.c
Project: CMakelists.txt x main.c
44     do {
45         printf("Podaj NORMALNA ilosc lat(wieksza od zera)\n");
46         scanf(" %d", &lata);
47     } while(lata<=0);
48 }
49 float lata_pom = lata*3.156;
50 int notacja = 7;
51 if(lata_pom>=10){
52     do {
53         lata_pom/=10;
54         notacja++;
55     }while(lata_pom>=10);
56 }
57 printf("Tvoj wiek w sekundach, wyrazony w notacji wykladniczej, to:\n%f*10^%d\n", lata_pom, notacja);
58 }
59
60 int main() {
61     zad1();
62     zad2();
63     zad3();
64     zad4();
65     return 0;
66 }
67
68 Version Control Run Python Packages TODO Messages CMake Problems Terminal Services
Build finished in 317 ms (3 minutes ago) clang-tidy 3.13 LF UTF-8 4 spaces C:PodsProgSpr1 | Debug
```

```
Run: PodsProgSpr1
C:\Users\koku\CLionProjects\PodsProgSpr1\cmake-build-debug\PodsProgSpr1.exe
Zadanie 1
Ustalona liczba(76), w formacie int, wyświetlona w formacie float z jednym i trzema miejscami po przecinku oraz w formacie napisowym:
76.0
76.000
76.000
Zadanie 2
Podaj kod ASCII aby otrzymać znak
Y
Twoj znak to: Y
Zadanie 3: obliczanie BMI
Podaj swoją wagę[kg] oraz wzrost[cm] według wzoru: masa, wzrost
70 170
Twoje BMI wynosi: 22.53
Zadanie 4
Podaj swój wiek w latach, aby otrzymać jego wartość w sekundach
60
Twoj wiek w sekundach, wyrażony w notacji wykładniczej, to:
6.312000e+08
Process finished with exit code 0
```

Wyjaśnienie napisanego przeze mnie kodu:

Zadanie 1:

Inicjalizuję zmienną `n = 76`. Aby wyświetlić ją dla użytkownika w formatach float z jednym i trzema przecinkami oraz napisowym, używam polecenia `printf`, w którym formatuje tekst wyświetlany przez program użytkownikowi, a następnie dokonuje konwersji zmiennej „`n`” (typu `int`) na wyżej wymienione typy, według kolejności formatowania.

Zadanie 2:

Definiuję zmienną „`a`”, aby móc użyć jej do przechowywania danych wprowadzonych przez użytkownika. Poleceniem `printf` informuję użytkownika o konieczności podania kodu ASCII w celu otrzymania przyporządkowanego mu znaku. Funkcją `scanf` pobieram od użytkownika dane. W przypadku podania przez użytkownika zbyt wielkiego, bądź zbyt małego kodu, co weryfikowane jest warunkiem `if` oraz pętlą `do while`, program prosi go o wprowadzenie poprawnej wartości. Następnie, poprawne kody są konwertowane na typ znakowy, w celu wyświetlenia poprawnego znaku poleceniem `printf`.

Zadanie 3:

Definiuję zmienne „`m`” i „`h`”, które posłużą mi do pobrania od użytkownika jego masy i wzrostu, za pomocą polecenia `scanf`, w celu obliczenia BMI. Następnie, program weryfikuje, za pomocą instrukcji `if` oraz pętli `do while`, czy użytkownik nie podał zbyt małej masy, bądź wzrostu i upomina go o podanie poprawnych wartości. Później, program konwertuje podany wzrost z centymetrów na metry oraz podnosi go do potęgi drugiej, zgodnie ze wzorem na BMI. Potem dokończa liczenie BMI, poprzez podzielenie zmiennej `m` (masy) przez zaktualizowaną zmienną `h` (wzrost). Na koniec, program wypisuje użytkownikowi komunikat: „Twoje BMI wynosi: [bmi]”, gdzie zmienna `bmi` jest liczbą typu float z dwoma miejscami po przecinku, w celu bardziej estetycznej prezentacji danych.

Zadanie 4:

Definiuję zmienną „`lata`”, która posłuży do pobrania od użytkownika jego wieku w latach, w celu przeliczenia go na sekundy. Za pomocą warunku `if` oraz pętli `do while`, program sprawdza czy

użytkownik nie podał niepoprawnego wieku oraz prosi go o wpisanie normalnej ilości lat. Poprawny wiek zostaje użyty przy inicjalizacji zmiennej „lata_pom”, która mnoży wiek użytkownika przez mantysę notacji wykładniczej, przedstawiającej ile mniej więcej znajduje się sekund w roku. Następnie, program inicjalizuje zmienną notacja, która przechowuje wykładnik notacji. Jeżeli „lata_pom” jest większe od 10, program ,za pomocą warunku if oraz pętli do while, dzieli zmienną „lata_pom” przez 10, jednocześnie zwiększając wykładnik potęgi o 1, po każdorazowym wykonaniu dzielenia. Po otrzymaniu poprawnej notacji wykładniczej wieku użytkownika w sekundach, program wypisuje ją. Do zapisania wieku w sekundach użyłem notacji wykładniczej, ze względu na fakt, iż program zwracał ogromne liczby, które były wypisywane jako ujemne.

Wnioski:

Ćwiczenie pomogło mi nieco zaznajomić się z językiem C. Poznałem parę sposobów pobierania i zwracania danych przez program, a także popracowałem nieco na pętli do while oraz dzieliłem program na funkcje. Niestety nie wiem jak zabezpieczyć program przed wprowadzaniem złego typu danych, np. charów tam, gdzie powinny być inty. Wiem natomiast jak weryfikować wielkość wprowadzanych danych.