C - 2 - Instrukcje warunkowe

Utwórz w Visualu nowy projekt, wykasuj od razu linijkę drukującą na ekranie napis **Hello World**.

2.1 Konstrukcja *if-else*

Z podstawową konstrukcją **if-else** zapoznaliśmy się już w poprzednim materiale.
Instrukcje warunkowe można zagnieżdżać, np:

```
int n=7;
if(n>0){
        if(n%2==0){
                printf("parzysta");
        }
        else if(n%3==0){
                printf("nieparzysta ale podzielna przez 3");
        }
        else{
                printf("jakaś inna ... ");
        }
}
else{
        printf("ujemna, nie chce mi sie prawdzac parzystosci");
}
```

Właśnie tu mamy zagnieżdżenie – nie zapomnij o klamerkach po zewnętrznym ifie !!!

Skoro nie zaszły dwa pierwsze żółte warunki, to liczba jest nieparzysta i niepodzielna przez 3.

Zadanie 0. Warunki logiczne (tutaj: **n>0**, **n%2==0** oraz **n%3==0**) przyjmują wartość **true** lub **false**. Generalnie, jeżeli w **if**ie umieścimy jakąkolwiek wartość różną od zera (nawet ujemną!), to będzie to uznane za **true**. Sprawdź to w jakimś krótkim programie.

Zadanie 1. Napisz program, w którym użytkownik poda z klawiatury swój wiek. Program ma za zadanie wyświetlić swój komentarz na temat wieku, np:

co najwyżej 5 lat	Przedszkolak. Odejdź od komputera !
więcej niż 5, ale mniej niż 16	Uczeń. Ucz sie, dziecko ....
co najmniej 16, ale mniej niż 65	Wiek produkcyjny. Produkuj, produkuj cokolwiek !
co najmniej 65 lat	No, to juz z gorki ... Że tez Ci sie jeszcze chce ...

Zadanie 2. Napisz program, w którym użytkownik poda z klawiatury liczbę całkowitą (rok czterocyfrowy). Program ma za zadanie odpowiedzieć na pytanie, czy podany rok jest rokiem przestępnym.

Przypomnienie: rok jest przestępny, jeżeli jest podzielny przez 4 i nie jest podzielny przez 100. Podzielność sprawdzamy za pomocą znaczka **%** (modulo).

Zadanie 3. Napisz program, w którym użytkownik poda z klawiatury liczbę całkowitą od 1 do 12. Jeżeli podano liczbę mniejszą od jeden lub większą od 12, to wyświetl komunikat o błędzie. W przeciwnym przypadku program powinien wyświetlić, do jakiej pory roku należy miesiąc o podanym numerze. W tym programie przydadzą się zagnieżdżone **if-else**y.

Przyjmujemy: zima: 12,1,2; wiosna: 3,4,5; lato: 6,7,8; zima: 9,10,11.

Zadanie 4. Napisz program, w którym użytkownik poda z klawiatury jakiś znaczek (literę, przecinek, cyfrę - cokolwiek), a program stwierdzi, czy wprowadzono literę dużą (kody ASCII od 65 dla **A** do 90 dla **Z**), literę małą (97 dla **a** aż do 122 dla **z**), cyfrę (48 dla **0** aż do 57 dla **9**), czy jakiś inny znak.

2.2 Operator warunkowy *? :*

Konstrukcję **if-else** można niesłychanie zagmatwać za pomocą operatora warunkowego. Jego składnia wygląda następująco:

[wyrażenie logiczne 1] ? [wyrażenie 2] : [wyrażenie 3]

Przykład:

```
int x,y,z;
y = 7;
z = 8;
x = (y>z) ? 3 : 4;
```

Ta linijka oznacza dokładnie to samo, co:

```
if(y > z){
    x = 3;
}
else{
    x = 4;
}
```

Taki operator warunkowy można też umieścić w **printf**ie:

```
printf( "Wieksza wartosc: %d", ((y > z) ? y : z) );
```

Ponieważ operator warunkowy jest dość skomplikowany dla początkowych programistów, będziemy z niego rzadko korzystać.

Zadanie 5. Zrób od nowa **zadanie 2** korzystając z operatora warunkowego.

Zadanie 6. Napisz program, który pobiera z klawiatury dwie liczby zmiennoprzecinkowe i orzeka, która z nich jest mniejsza. Skorzystaj z operatora warunkowego.

2.3 Konstrukcja *switch*

Jeżeli mamy do czynienia z mnóstwem możliwości, to zamiast **if-else** skorzystamy z konstrukcji **swich**.
Przykład:

```
...
int liczba;
printf("Podaj liczbę od 1 do 5");
scanf("%d", &liczba);

 switch (liczba){
        case 1:{
                printf("Podales rzymskie I.");
                break;
               }
        case 2:{
                printf("Podales rzymskie II.");
                break;
               }
        case 3:{
                printf("Podales rzymskie III.");
                break;
               }
        case 4:{
                printf("Podales rzymskie IV.");
                break;
               }
        case 5:{
                printf("Podales rzymskie V.");
                break;
               }
        default:{
                printf("Podales cos dziwnego ...");
                break;
                }
 }
...
```

break (ang. przerwij) daje programowi do zrozumienia, że znaleźliśmy przypadek dla naszej liczby i nie trzeba szukać dalej.

Bez **break**a program szukałby dalej – jeżeli poniżej byłby przypadek który uwzględnia też naszą liczbę, to mogłoby to doprowadzić do błędnego działania (chociaż zależy to od tego, co chcemy z tą liczbą zrobić)

Instrukcje w sekcji default wykonują się, gdy nasza liczba nie wpadnie w żaden z wcześniejszych case'ów.

Zadanie 7. Napisz program, który pobiera z klawiatury literkę (małą lub dużą). Program ma za zadanie wypisać nazwy miesięcy zaczynających się na podaną literkę - skorzystaj z konstrukcji **switch**. Użyj sekcji **default** dla obsłużenia literek, na które nie zaczynają się żadne miesiące lub dla znaków, które nie są literkami (nie wolno podejrzewać użytkownika o to, że zawsze będzie podawał poprawne dane).

Wskazówka: jeżeli chodzi o małe i duże literki, to dozwolone jest coś takiego:

```
....
case 's' :
case 'S' :{
        printf("Styczeń, sierpien.");
        break;
   }
....
```

Nie trzeba więc kopiować tej samej zawartości **case**a dla dużych literek.

Zadanie 8. Napisz program, który wyświetli menu:

0) Wyjscie z programu.
1) Rownanie liniowe
2) Rownanie kwadratowe
Podaj opcje:

Pobieraj opcję jako znak typu **char**.
Jeżeli podano coś innego, niż 0, 1, 2, to wyświetl komunikat „nieznana opcja" i nie rób nic więcej.
Jeżeli podano 0 – wypisz „do widzenia" i nie rób nic więcej.

Jeżeli podano 1 – poproś o współczynniki A oraz B, wyświetl informację o pierwiastku, braku rozwiązań lub o nieskończonej liczbie rozwiązań.
Jeżeli podano 2 – poproś o współczynniki A, B oraz C, oblicz deltę i wyświetl wartości pierwiastków lub informację o ich braku.
Skorzystaj przynajmniej raz z konstrukcji **switch**. Wewnątrz **case**ów możesz korzystać z konstrukcji if-else do zbadania liczby pierwiastków.

2.4 Najczęstsze błędy

Używanie jednego znaku równości do porównywania wartości w **if**ie.
Nie otaczanie znaków typu **char** apostrofami (patrz: zadania 4, 7, oraz 8).
Zapominanie o **break**ach przy okazji **case**ów.
Gubienie dwukropków po **case**ach.
Pisanie średnika zaraz po **if**ie, np: **if(x==2) ;** - oznacza to tyle, co: „jeżeli **x** wynosi 2, to nic nie rób".

2.5 Większy program - kalkulator

Napisz program, który przyjmuje z klawiatury 3 dane: liczbę, znak i znowu liczbę (a więc w **scanf**ie będzie %lf %c %lf).
- jeżeli znakiem jest + - *, to wykonaj na pobranych liczbach stosowne działanie i zwróć wynik.
- jeżeli znakiem jest /, to sprawdź, czy druga liczba nie jest zerem i w miarę możliwości wykonaj dzielenie lub wyświetl komunikat o błędzie.
- jeżeli podano znak ^, to wykonaj potęgowanie – służy do tego funkcja **pow** z biblioteki **math.h**. Funkcja ta przyjmuje dwa argumenty typu **double** oddzielone przecinkiem – pierwszy to liczba do spotęgowania, a drugi to potęga. Przykład użycia:

```
printf("2.5 do 3.1 wynosi %lf \n",pow(2.5,3.1));
```

lub:

```
double wynik = pow(2.5,3.1);
printf("2.5 do 3.1 wynosi %lf \n",wynik);
```

- jeżeli podano %, to wyświetl resztę z dzielenia modulo – pamiętaj, by rzutować liczby na typ **int**. Ewentualnie, przypisz swoje liczby do nowych zmiennych typu **int** i na tych nowych zmiennych oblicz resztę z dzielenia modulo.
- jeżeli podano jakiś inny znak, to wyświetl komunikat o błędzie.

Do zbadania znaku operacji +,-,*,/,%,^ skorzystaj z konstrukcji **switch**.

2.6 Quiz

1. Jaka będzie wartość **x** na końcu poniższego kodu i dlaczego?

```
int x=5,y=10;
y += x/2;
x = y%5;
```

_____
_____

2. Połącz dwa **if**y w jeden:

```
if (x > 1){
        if ( x < 10 ){
                prinf("%d należy do (1,10)",x);
        }
}
```

_____
_____
_____

3. Wiedząc, że **x=2**, **y=6** i **z=2** orzeknij, jaką wartość zwróci każdy **if** – **true**, czy **false**?

```
if(x==4)    _____
if(x!=y-z)  _____
if(z=1)     _____ (*)
if(y)       _____ (*)
```

4. Co zostanie wypisane na ekranie i dlaczego jest to niezależne od wartości **x**? Popraw błąd na czerwono.

```
if ( x < 10 );
        printf("A ku ku !!!");
```

_____
_____