

ISTOTNE UWAGI DO PONIŻSZYCH ZADAŃ:

- Należy przyjąć zasadę: jeśli plik (kod źródłowy) programu nazywa się: **zad1.c**, to po kompilacji plik wykonywalny nosi nazwę: **zad1** (i podobnie dla innych plików).
- Kod winien być pisany przejrzysto z uwzględnieniem wcięć podkreślających poziom polecenia;
- Wszystkie zadania proszę umieszczać w jednym wspólnym katalogu o nazwie: **MITP**

Zad. 1.

Należy napisać program (**zad1.c**), którego rezultatem jest pojawiający się na ekranie tekst jak po prawej:

UWAGA: Proszę użyć znaków specjalnych dla nowego akapitu i tabulacji.

```
%%%%%%%%%%  
      Witaj!  
%%%%%%%%%%  
  
po pojedynczym  
akapicie  
  
pojedyncza tabulacja
```

Zad. 2.

Należy napisać program (**zad2.c**), który pobiera od użytkownika dwie dowolne liczby (rzeczywiste) i wypisuje która z nich jest większa, bądź że są sobie równe.

Zad. 3.

Należy napisać program (**zad3.c**), który przelicza temperaturę podaną w stopniach Celsjusza na podaną w stopniach Fahrenheita.

UWAGA:**0° C = 32° F****100° C = 212° F****Zad. 4.**

Należy napisać program (**zad4.c**), który sprawdza, czy pobrana od użytkownika dowolna liczba jest całkowita, czy nie, a jeśli jest to czy dodatkowo jest parzysta, czy nieparzysta.

Zad. 5.

Należy napisać program (**zad5.c**), wyliczający wartość podanego wyrażenia zawierającego dokładnie dwie wartości liczbowe (całkowite) i znak operatora (tylko dla: *, -, +, /)

Przykładowo:

```
podaj wyrażenie: 12+5  
wynik: 17
```

Zad. 6.

Należy napisać program (**zad6.c**), pobierający od użytkownika cztery dowolne liczby całkowite i wypisujących ich sumę w podobnym stylu jak obok:

UWAGA: Zakładamy, że liczba pobierana od użytkownika składa się z do 4 cyfr a wynik może być złożony z 5 cyfr.

```
123  
1000  
  1  
 17  
----  
1141
```

Zad. 7.

Należy napisać program (**zad7.c**), pobierający od użytkownika dowolny znak i wypisujący odpowiadający mu numer w kodzie ASCII.

Zagadnienia do przygotowania na najbliższe zajęcia:

- wyrażenia warunkowe: **switch**;
- pętle: **for**, **while** i **do-while**;
- polecenia sterujące **break** i **continue**;
- instrukcja skoku: **goto**
- etykiety;
- zaawansowana znajomość funkcji wejścia-wyjścia **printf** i **scanf**, a w szczególności rola wartości liczbowych specyfikujących pole (np. **%4.1f**, **%*d**, etc.)
- priorytety podstawowych operatorów (*jednoargumentowe*: **++**, **--**, **sizeof**, **(typ)**, **-**, **+**, **!**, **~**, *dwuargumentowych*: **+**, **-**, **/**, **%**, **&&**, **||**, **&**, **|**, **<<**, **>>**, **<**, **<=**, **>**, **>=**, **!=**, **==**, **=**, **+=**, **-=**, **/=**, **%=**), oraz ich *tętność* (prawo- lewostronna);