



# **Uniwersytet Jana Długosza w Częstochowie**

Mykhailo Hulii

Informatyka, 2 rok,

Studia 1 stopnia

Grupa 2

## Zadanie 4

Napisać i użyć w programie funkcję o nazwie bcdOK sprawdzającą czy podana, jako argument, wartość jest liczbą szesnastkową BCD. Przyjmujemy zakres poprawnych znaków: 0..F. Przykładowe wywołanie funkcji: bcdOK("absgdfsr");

```
Task4!  
Czy 'absgdfsr' jest liczba szesnastkowa BCD? Nie  
Czy '1234ABCF' jest liczba szesnastkowa BCD? Tak
```

## Zadanie 5

Napisz funkcję swap(a,b) zamieniającą jej argumenty czyli a z b. Zastanowić się jak zwrócić te argumenty do programu. Dodatkowo, jeśli a jest równe b funkcja zwraca 0, jeśli a>b zwraca 1 a jeśli a<b zwraca -1;

```
Task5!  
Przed zamianą: x = 5, y = 3  
Po zamianie: x = 3, y = 5  
Porównanie a i b: 0  
Porównanie a i b: 1  
Porównanie a i b: -1
```

## Zadanie 6

Napisz i użyj w programie funkcję podnoszącą jej argument do kwadratu. Napisz dwie wersje funkcji, pierwsza typu long int pobiera argument przez wartość druga "typu" void pobiera i zwraca argument przez referencję.

```
Task6!  
Liczba 5 podniesiona do kwadratu wynosi: 25
```

## Zadanie 6a

Napisz funkcję pozwalającą wyznaczyć adres sieci z podanego jako argument adresu IP. Do przechowywania danych użyj struktury.

```
Task6a!  
Adres IP: 192.168.1.10  
Maska sieci: 255.255.255.0  
Adres sieci: 192.168.1.0
```

## Zadanie 7

Napisz kod trzech funkcji przeciążonych

- pierwsza funkcja liczy pole kwadratu i przyjmuje jeden parametr całkowity
- druga funkcja liczy pole prostokąta i przyjmuje dwa parametry rzeczywiste
- trzecia funkcja liczy objętość sześcianu i przyjmuje trzy parametry rzeczywiste z których dwa mają zdefiniowane wartości domyślne.

W programie głównym stwórz menu, dające użytkownikowi możliwość wyboru tego co chce policzyć. W zależności od wybranej opcji program ma zapytać o parametry i podać wynik, używając do tego stosownej funkcji.

```
Task7!  
Choose an option:  
1. Area of a square  
2. Area of a rectangle  
3. Volume of a cube  
Your choice: 1  
Enter the side length of the square: 12  
The area of the square is: 144
```

```
Task7!  
Choose an option:  
1. Area of a square  
2. Area of a rectangle  
3. Volume of a cube  
Your choice: 2  
Enter the length of the rectangle: 12  
Enter the width of the rectangle: 23  
The area of the rectangle is: 276
```

```
Task7!  
Choose an option:  
1. Area of a square  
2. Area of a rectangle  
3. Volume of a cube  
Your choice: 3  
Enter the length of the cube: 12  
Enter the width of the cube (default is 1): 23  
Enter the height of the cube (default is 1): 10  
The volume of the cube is: 2760
```

## Zadanie 8

Napisać funkcję *pprostok*, która otrzymuje dwa parametry - długości dwóch boków trójkąta prostokątnego (przyprostokątne), a zwraca wartość - długość przeciwprostokątnej  $c^2=a^2+b^2$ . Jeśli któryś z argumentów wynosi zero lub jest ujemny funkcja ma zwracać wartość -1. Funkcję wykorzystać w programie, zadającym użytkownikowi pytanie o podanie długości przyprostokątnych, jeśli parametry są bez sensu (zero lub ujemne) dzięki zwracaniu wartości -1 przez funkcję należy wyświetlić odpowiedni komunikat. W programie wykorzystać napisaną przez siebie funkcję *kwadrat* podnoszącą do kwadratu zadany argument

```
Task8!
Enter the length of the first side of the right triangle: 12
Enter the length of the second side of the right triangle: 23
The length of the hypotenuse is: 25.9422
```

## Zadanie 8a

Utworzyć funkcję otrzymującą jako argumenty tablicę oraz wartość. Funkcja szuka podanej wartości w tablicy i zwraca jej indeks. Jeśli elementu brak to zwraca wartość -1. W jaki sposób w funkcji można "wybadać" ilość elementów tablicy?

```
Task8a!  
Value 7 found at index 2
```

## Zadanie 10

Napisz dwie funkcje: `ustawBit` i `kasujBit`. Funkcja przyjmuje trzy argumenty: słowo do zmiany bitu, nr bitu do ustawienia lub kasowania, oraz informację co ma być wykonane ustawienie czy kasowanie. Funkcja zwraca zmienną `słowo` przez referencję oraz kod wykonania operacji: 0 - ok, -1 - słowo jest <0, -2 słowo przekracza dopuszczalny zakres, -3 - bit <0 itd. Wykorzystaj maski bitowe zdefiniowane w typie `wyliczeniowym`. (Maski możesz wykorzystać z zadania Moduł 4 Zadanie 12). Jako pomoc możesz napisać dodatkową funkcję zamieniającą i wyświetlającą wartość dziesiętną na binarną (np funkcję `DEC2BIN`).

```
Task10!  
Original value in binary: 000000000000000000000000000001101  
After setting bit 2, value in binary: 000000000000000000000000000001101  
After clearing bit 3, value in binary: 00000000000000000000000000000101
```

## Zadanie 11

Utwórz funkcję przyjmującą dowolną ilość elementów (funkcja z nieokreśloną ilością argumentów) ciągu liczb naturalnych. Elementem końcowym jest zero. W wyniku działania funkcja ma podać: ilość elementów ciągu, (zero nie wchodzi w skład elementów ciągu), sumę tych elementów, ich iloczyn, wartość maksymalną i minimalną.

---

## Zadanie 12

Napisać funkcję przyjmującą jako argumenty tablicę liczb *t* oraz liczbę *n*. Funkcja w tablicy *t* wyszukuje wartości *n* i zwraca ich ilość. Rozważyć przekazanie do funkcji tablicy bez podanego wymiaru oraz problem określenia tego wymiaru w funkcji. Jako pomoc napisać funkcję *lostab* wypełniającą podaną, jako argument, tablicę wartościami losowymi z zakresu 0..*n* gdzie *n* jest jednym z argumentów tejże funkcji.

```
Task12!  
Liczba wystąpień liczby 5 w tablicy: 2
```

## Zadanie 15

Napisz funkcję z trzema parametrami: *i*, *tablica*, *n*. Funkcja zwraca wartość *i*-tego elementu tablicy, oraz wartość *n* będącą ilością elementów w tablicy różnych od zera.

```
Task15!  
Value of the 2-th element of the array (non-zero): 5
```

## Zadanie 16

Napisz funkcję zamieniającą pobraną wartość ilości sekund na dni, godziny, minuty i sekundy. Rozważ jak funkcja może zwracać te kilka wartości i jakiego typu powinna ona być. Użyj jej w programie wczytującym od użytkownika ilość sekund o wyświetlającym coś w rodzaju:

```
Podaj sekundy [0 konczy]: 20000
20000 sekund to jest 0 dni 5 godz 33 min 20 sek
lub
dd:gg:mm:ss 0:5:33:20
Podaj sekundy [0 konczy]:
```

```
Task16!
Enter the number of seconds: 654324
654324 seconds is equivalent to:
Days: 7
Hours: 13
Minutes: 45
Seconds: 24
```

## Zadanie 18

Napisać funkcję `liczBity` zliczającą bity ustawiona na jeden w podanym argumencie

```
Task18!
Enter a number: 123
Number of bits set to 1 in 123: 6
```

## Zadanie 19

Napisać funkcję obliczającą długość efektywną podanego ciągu znaków.

```
Task19!
Enter a string: Hello,World!
Effective length of the string: 12
```

## Zadanie 20

Napisać funkcję sprawdzającą czy podany rok jest rokiem przestępnym. Jeśli tak funkcja zwraca 1 jeśli nie to zwraca 0;

```
Task20!
Enter a year: 2050
Is 2050 a leap year? 0
```

## Zadanie 21

Napisać funkcję zamieniającą liczbę szesnastkową na dziesiętną. Liczba szesnastkowa zaczyna się od 0x lub 0X.

```
Task21!  
Enter a hexadecimal number (starting with 0x or 0X): 0x7B  
Decimal equivalent of 0x7B is: 123
```

## Zadanie 24

Napisać program z funkcją przyjmującą cztery parametry: dwa pierwsze typ int, kolejne dwa to wskaźniki typu int. Funkcja ma zwracać:

- iloraz dwóch pierwszych parametrów
- za pomocą trzeciego parametry sumę dwóch pierwszych
- za pomocą czwartego parametry iloczyn dwóch pierwszych

```
Task24!  
Enter two numbers: 12 23  
Sum of 12 and 23 is: 35  
Product of 12 and 23 is: 276  
Quotient of 12 and 23 is: 0.521739
```

## Zadanie 27

Napisać funkcję symetria poszukującą w dostarczonej jej tablicy pierwszego wystąpienia dwóch identycznych wartości ułożonych symetrycznie względem środka tablicy.. Funkcja jako argumenty przyjmuje: tablicę wartości całkowitych, rozmiar tablicy. Funkcja zwraca przez wskaźnik na zmienną wartość pierwszej i ostatniej komórki tablicy, przez wartość informację o tym czy znalazła symetrię czy nie, oraz przez referencję nr komórki i jej zawartość elementu symetrycznego o ile taki istnieje. Jeśli symetrii w tablicy brak zwracana wartość nr komórki z symetrią ma wynosić -1.

```
Task27!  
Symmetry found!  
First value: 1 at index 0  
Last value: 1 at index 7  
Symmetric index: 0 with value: 1
```

## Druga część

### Zadanie 9a

Napisać program obrazujący kwestię przekazania do funkcji argumentu jako wskaźnika. Funkcja powinna "spróbować" zmienić to na co dostarczony wskaźnik wskazuje (zwrot danych przez argument) oraz zawartość samego wskaźnika czyli adres na jaki oryginalny wskaźnik wskazuje. Omówić kwestię przekazania danych jako wskaźników do funkcji i problemy z tym związane.

```
Task9a!  
Before calling modify function:  
Value at ptr: 42  
Address stored in ptr: 0x7fff9179d18c  
  
After calling modify function:  
Value at ptr: 100  
Address stored in ptr: 0x7fff9179d18c  
  
Value of 'number' variable outside function: 100
```

### Zadanie 5a

Napisać i omówić działanie funkcji otrzymującej wskaźnik do tekstu a zwracającej jego długość. Zwrócić uwagę na to w jaki sposób dane są przekazywane do funkcji.

```
Task5a!  
Length of the text "Hello, World!" is: 13
```



## Zadanie 13

Napisać funkcję przyjmującą dwie zmienne wskaźnikowe. Funkcja ma wymienić wartości tych zmiennych między sobą. Napisać drugą funkcję realizującą to samo ale przyjmującą zmienne referencyjne.

Napisać funkcję (i użyć jej w przykładowym programie) przyjmującą dwa argumenty: wskaźnik na zmienną typu int oraz zmienną typu int przez referencję. Funkcja ma zamienić obie wartości.

```
Task13!
Before swapping using pointers:
num1 = 10, num2 = 20
After swapping using pointers:
num1 = 20, num2 = 10

Resetting values for swapping using references:
Before swapping using references:
num1 = 10, num2 = 20
After swapping using references:
num1 = 20, num2 = 10

Resetting values for swapping pointer and reference:
Before swapping pointer and reference:
num1 = 10, num2 = 20
After swapping pointer and reference:
num1 = 20, num2 = 10
```

## Zadanie 14

Napisać funkcje przeciążone "wskret". Pierwsza funkcja nie posiada argumentów, alokuje pamięć dla liczby typu int i zwraca jej wskaźnik. Druga funkcja posiada jeden argument typu int, jest to liczba jaka ma być wpisana do utworzonej zmiennej dynamicznej jak wyżej. Ta funkcja także zwraca wskaźnik do tej zmiennej.

```
Task14!
Value allocated using allocateMemory(): 42
Value allocated and initialized using allocateAndInitialize(): 99
```

## Zadanie 15

Napisać funkcję przyjmującą jako argument wartość całkowitą "n". Funkcja rezerwuje w pamięci blok zmiennych typu int o ilości elementów równych "n" a zwraca adres pierwszego elementu tego bloku. Funkcja wypełnia tak utworzony blok pamięci wartościami całkowitymi od 1 do n. W programie głównym, po wywołaniu funkcji odczytać i wyświetlić zawartość elementów tego bloku pamięci. Zadanie napisać w C i w C++ celem zobrazowanie różnic w zarządzaniu pamięcią.

```
Task15_2!  
Enter the number of integers to allocate: 8  
Contents of the allocated block:  
1 2 3 4 5 6 7 8
```

## Zadanie 16

Napisz funkcję otrzymującą jako argumenty: wskaźnik na stałą oraz wskaźnik na zmienną. Funkcja ma za zadanie przepisać wartość stałej do zmiennej. Napisz drugą funkcję otrzymującą jako argumenty: wskaźnik na stałą oraz stały wskaźnik na zmienną. Funkcja ma za zadanie przepisać wartość stałej do zmiennej. W programie przeprowadź możliwość aktualizacja stałego wskaźnika oraz zmiany danych wskaźnika na zmienną.

```
Task16_2!  
Before calling copyConstantToVariable function:  
constantValue: 10  
variableValue: 0  
  
After calling copyConstantToVariable function:  
constantValue: 10  
variableValue: 10  
  
After updating constantValue:  
constantValue: 20  
variableValue: 10  
  
After changing ptrToVariable to point to newValue:  
constantValue: 20  
variableValue: 10
```

## Zadanie 20

Napisz dwie funkcje operujące na tablicach. Pierwsza funkcja przyjmuje ilość elementów tablicy, tablicę (przez operator indeksowania) i wypełnia tablice kolejnymi liczbami naturalnymi od zero. Druga funkcja przyjmuje jako argumenty dwie tablice (przez wskaźnik) i liczbę elementów każdej z nich (są identyczne) i przepisuje elementy tablicy pierwszej do drugiej. Napisać także funkcję pokaz wyświetlającą tablice (argumenty n i tablica), w tej funkcji zrobić przegląd różnych sposobów dostarczania do funkcji tablicy oraz adresowania jej komórek.

```
Task20_2!  
Array elements:  
0 1 2 3 4  
Array elements:  
0 1 2 3 4
```

## Zadanie 21

Napisać funkcję przesuwającą elementy w tablicy o podaną jako argument wartość "ile" w lewo lub w prawo (argument "kierunek"). Możesz użyć tylko jednej tablicy. Argumenty "ile" oraz "kierunek" mają mieć nadane wartości domyślne. Wartości, które "wypadną" z tablicy z prawej mają wejść do niej z lewej i wstak.

```
Task21_2!  
Array elements:  
1 2 3 4 5  
Array elements:  
5 1 2 3 4  
Array elements:  
1 2 3 4 5  
Array elements:  
4 5 1 2 3
```