
PROGRAMOWANIE IMPERATYWNE

ZAD. 1. Uruchom i przeanalizuj poniższy program.

```
#include <stdio.h>
int main(void){
    printf("Hello, World!\n");
    return 0;
}
```

A) Przekształć powyższy program w taki sposób aby wyświetlał Twoje imię i nazwisko w jednej linii oraz indeks w kolejnej.

ZAD. 2. Uruchom i przeanalizuj poniższy program.

```
#include <stdio.h>
int main(void){
    printf("%d\n", 50/13*13 + 50%13);
    return 0;
}
```

A) Wypróbuj inne warianty operatora %d. Opisz swoje wyniki.

* <https://www.cplusplus.com/reference/cstdio/printf/>

ZAD. 3. Uruchom i przeanalizuj poniższy program.

```
#include <stdio.h>
int main(void){
    int x;
    scanf("%d", &x);
    printf("%d\n", x/13*13 + x%13);
    return 0;
}
```

A) Przekształć powyższy program w taki sposób by pobierał od użytkownika wszystkie liczby (tzn. Przedstaw wartość 13 w postaci zmiennej).

ZAD. 4. Uruchom i przeanalizuj poniższy program.

```
#include <stdio.h>
int main(void){
    int x;
    scanf("%d", &x);
    if( x < 0 ) {
        x = -x;
    }
}
```

```

    }
    printf("%d\n", x);
    return 0;
}

```

A) Co powoduje błąd kompilacji? W jaki sposób rozwiązać ten problem?

ZAD. 5. Napisać w C program odpowiadający na pytanie (Tak/Nie), czy podane przez użytkownika trzy liczby naturalne mogą być bokami trójkąta równoramiennego/równobocznego?

ZAD. 6. Uruchom i przeanalizuj poniższy program.

```

#include <stdio.h>
int digi(int a){
    int r, p;
    r = 1;
    p = 10;
    while(a >= p) {
        r = r + 1;
        p = p * 10;
    }
    return r;
}

int main(){
    int x;
    scanf("%d", &x);
    printf("%d\n", digi(x));
    return 0;
}

```

A) Przekształć powyższy program w taki sposób, aby zwracał liczbę cyfr binarnych (0,1) potrzebnych do zapisania podanej liczby naturalnej x .

B) Uogólnij funkcję *digi* wprowadzając drugi parametr b (oznaczający „base”), aby zwracała liczbę cyfr potrzebnych do przedstawienia liczby naturalnej x w systemie o dowolnej podstawie b .

ZAD. 7. Uruchom i przeanalizuj poniższy program.

```

#include <stdio.h>
int main(void){
    int n, x[10], i;
    scanf("%d", &n);
    i=0;
    while(i<n){

```

```

        scanf("%d", &x[i]);
        i=i+1;
    }
    i=i-1;
    while(i>=0){
        printf("%d", x[i]);
        i= i-1;
    }
    printf("\n");
    return 0;
}

```

A) Przekształć powyższy program w taki sposób, aby zwracał sumę liczb podanych na wejściu.

B) Przekształć program z punktu A) w taki sposób, aby funkcjonalność została zachowana pomimo wyeliminowania tablicy x z zastosowanej implementacji.

ZAD. 8. Uruchom i przeanalizuj poniższy program.

```

:-initialization(main).
father(elizabeth2, george6).
father(george6, george5).
father(george5, edward7).
father(edward7, albertCoburg).
father(aliceHesse, albertCoburg).
mother(elizabeth2, elizabethBowesLyon).
mother(george6, maryTeck).
mother(george5, alexandraDenmark).
mother(edward7, queenVictoria).
mother(aliceHesse, queenVictoria).
parent(X,Y):-father(X,Y); mother(X,Y).
grandma(X,Y):-parent(X,Z), mother(Z,Y).
grandpa(X,Y):-parent(X,Z), father(Z,Y).
greatGrandma(X,Y):-parent(X,Z), grandma(Z,Y).
greatGrandpa(X,Y):-parent(X,Z), grandpa(Z,Y).
great2Grandma(X,Y):-parent(X,Z), greatGrandma(Z,Y).
great2Grandpa(X,Y):-parent(X,Z), greatGrandpa(Z,Y).
main:-great2Grandma(elizabeth2,Y),write(Y), nl.

```

A) Zmodyfikować pytanie (tzn. goal0) w taki sposób, aby dowiedzieć się kto jest pra-pra wnuczką królowej Wiktorii?