

Podstawy programowania, ćw. – lista nr 3.

Zad. 1. Zapisz poniższy program podany w wersji iteracyjnej z pętlą for generujący n kolejnych liczb ciągu Fibonacciego z zastosowaniem pętli a) do i b) while c) narysuj schematy blokowe.

```
#include <stdio.h>
main()
{ //liczba liczb ciągu
  const unsigned int n=9;
  unsigned int a, b, temp;
  a = 0; b = 1;
  for(unsigned int i=0; i < n; i++) {
    printf("%d, ", a);
    a += b;
    temp = a; a = b; b = temp;
  }
}
```

```
#include <stdio.h>
main()
{ const unsigned int n=9;
  unsigned int a, b, temp;
  a = 0; b = 1;
```

Zad. 2. Proszę napisać program rezerwujący miejsca dla pasażerów w 8 miejscowym minibusie. Zaproponuj strukturę danych do przechowywania informacji.

Zad. 3. Proszę napisać program rezerwujący miejsca dla pasażerów w kilku minibusach.

Zad. 4. Na czym polega różnica w działaniu obu funkcji do obliczania funkcji silnia. Narysuj schematy blokowe programu głównego i obu funkcji.

```
#include <stdio.h>

long int silnia1(int n){
  long int silnia = 1;

  for (int i=1; i<=n; i++)
    silnia=silnia*i;
  return silnia;
}

long int silnia2(int n){
  if (n==1) return 1;
  else return n*silnia2(n-1);
}

main()
{ printf("silnia= %ld\n",silnia1(4));
  printf("silnia= %ld",silnia2(4));
}
```

Uwaga: Użyj bloku „Proces uprzednio zdefiniowany”:

--	--	--

Zad. 5. Dlaczego poniższy program nie będzie w każdym wypadu dawał dobrych wyników?

```
main()
{ for (int i=1; i<=20; i++)
  printf("%2d!= %10ld\n",i,silnia2(i));
}
```

Zad. 6. Napisz program na rekurencyjne generowanie liczb ciągu Fibonacciego.

Zad. 7. Proszę uzupełnić kody programu. Definicja tablicy **char** `tablica[rozmiar];`

Lp.	Schemat blokowy	Kod programu
1.	Wprowadź zmienną <i>zm</i>	
2.	Wyświetl zmienną <i>zm</i> Czekaj na naciśnięcie klawisza.	
3.	<pre> graph TD A[Procedura] --> B[/Wyświetl „Czy ponownie uruchomić program? t/n”/] B --> C[/Wczytaj (decyzję użytkownika) zmienną opcja/] C --> D{Czy opcja = T lub t?} D -- T --> C D -- N --> E[] </pre>	
4.	Zapisz do tablicy zmienną <i>zm</i> na pozycji <i>i</i>	
5.	Zapisz do tablicy zmienną <i>zm</i> na pozycji <i>i</i> Ze sprawdzeniem, czy <i>i</i> nie przekracza stałej <i>rozmiar</i>	
6.	Pokaż element na pozycji <i>i</i>	
7.	Wypełnij tablicę znakiem O	
8.	Wyświetl wszystkie elementy tablicy różne od 'n' w postaci: nr_pozycji : wartość elementu	