

# **Podstawy Informatyki**

## **Ćwiczenia tablicowe Nr 5**

**Dr inż. Iwona Oprządkiewicz**

## **Plan dzisiejszych zajęć:**

- 1.Kartkówka z tablic**
- 2.Programy zawierające funkcje**
- 3.Tematyka następnych zajęć**

# Przekazywanie parametrów w C odbywa się przez wartość:

//.....

```
int main()
{
  typ_p1 a; typ_p2 b;
  typ1_p1 c; typ1_p2 d;
  typ_F0 f0 A; typ_F1 f1 B;
  //.....
  A=F0(a,b);// uruchomienie f-cji F0
  /*do zmiennej A podstawiana jest
  wartość zwracana przez f-cję F0*/
  //.....
  B=F1(c,d);// uruchomienie f-cji F1
  //.....
  return 0;
}

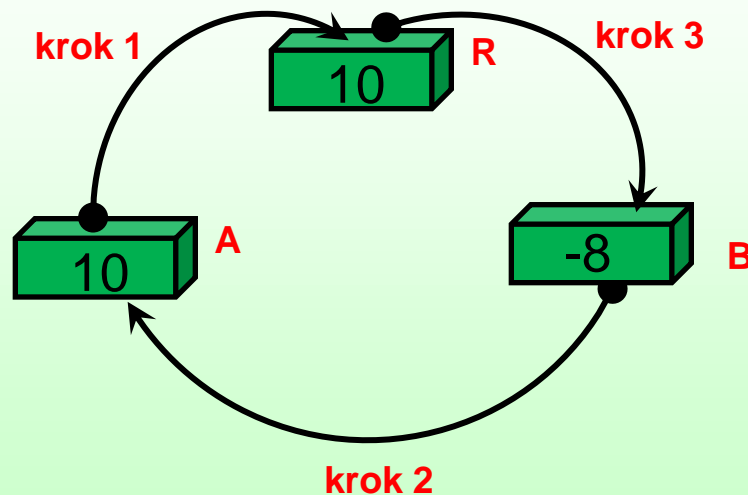
//.....
```

```
typ_F0 F0(p1,p2)
typ_p1 p1;
typ_p2 p2;
{typ_F0 x;
//.....
return x;
}
```

1. Tworzone są lokalne kopie argumentów a i b jako p1 i p2 (p1 ma wartość a, p2 wartość b).
2. Wykonywane są wszystkie operacje na zmiennych lokalnych (i globalnych, jeżeli takie występują i nie są przesłonięte).
3. return x – przekazuje wartość zmiennej lokalnej x do procedury wywołującej.
4. Pamięć zajęta przez wszystkie zmienne lokalne jest zwalniana

**Co zrobić, jeżeli nie potrzebujemy funkcji, która oblicza tylko jedną wartość, ale chcemy, żeby zwróciła do procedury wywołującej wartości więcej niż jednej zmiennej?**

Np. chcemy napisać procedurę, która zamienia wartości w dwóch zmiennych.



**Rozwiązanie I:**

**Wszystkie operacje w procedurze przeprowadzamy na zmiennych globalnych.**

**//Zmienne A i B są to zmienne globalne**

**void zamiana1()**

**{**

**int R;**

**R=A;**

**A=B;**

**B=R;**

**}**

**Wady:**

**Procedura nie jest uniwersalna (dla innych zmiennych musimy napisać następną procedurę)**

## Rozwiązanie II:

Procedura powinna mieć dwa parametry (formalne), które w momencie jej uruchomienia przyjmowałyby wartości wskazanych zmiennych (parametrów aktualnych).

//Zmienne A i B są to zmienne lokalne w procedurze wywołującej procedurę zamiana2

```
void zamiana2(z1,z2)
```

```
int z1;int z2;
```

```
{
```

```
int R;
```

```
R=z1;
```

```
z1=z2;
```

```
z2=R;
```

```
}
```

## Wady:

- zamiana wartości zmiennych nie jest widoczna w procedurze wywołującej funkcję

//wywołanie procedury zamiana2

```
int main()
```

```
{
```

```
int A=10;
```

```
int B=-8;
```

```
.....
```

```
zamiana2(A,B);
```

```
/*A nadal ma wartość 10, a  
B wartość -8 */
```

```
.....
```

```
}
```

### Rozwiązanie III:

Procedura powinna mieć dwa parametry (formalne), które w momencie jej uruchomienia przyjmowałyby wartości wskazanych zmiennych (parametrów aktualnych).

//Zmienne A i B są to zmienne lokalne w procedurze wywołującej procedurę zamiana2

```
void zamiana2(z1,z2)
```

```
int *z1, *z2;
```

```
{
```

```
int R;
```

```
R=*z1;
```

```
*z1=*z2;
```

```
*z2=R;
```

```
}
```

//wywołanie procedury zamiana2

```
int main()
```

```
{
```

```
int A=10;
```

```
int B=-8;
```

```
.....
```

```
zamiana2(&A,&B);
```

```
/*A ma teraz wartość -8, a  
B wartość 10 */
```

```
.....
```

```
}
```

W tym rozwiązaniu cel został osiągnięty



# Co zrobić, jeżeli chcemy jako parametr przekazać do procedury tablicę?

W takiej procedurze parametrem powinien być wskaźnik do zmiennej takiego samego typu, jak elementy tablicy, którą chcemy przekazać do procedury

Np.

```
int main()
{float liczby[10];
int N=5;
//.....
//procedura, która wczytuje tablicę
Czytaj (liczby,N);
//.....
return 0;
}
```

```
void Czytaj(x,m)
float *x;int m;
{
int k;
for(k=0;k<m;k++)
scanf(„%f”,x+k);
return 0;
}
```

# Tematy z funkcji:



1. Napisz program zawierający funkcję, która oblicza symbol Newtona . Program powinien wykonać obliczenia dla 10 zestawów danych ( $n$  i  $k$ ), wczytywanych z podanego przy uruchomieniu pliku (argument programu). Dla każdego zestawu należy wyświetlić liczby i wynik dwumianu.
- 2 . Napisz program, który zawiera funkcję obliczającą średnią arytmetyczną liczb zmiennoprzecinkowych z podanej tablicy (parametr funkcji). W programie należy wykonać funkcję kilkakrotnie dla różnych tablic.
3. Napisz program, który umożliwi wariantowe i wielokrotne wykonywanie następujących kilku funkcji: obliczanie pola prostokąta, obliczanie pola koła, obliczanie pola trójkąta. Program powinien kończyć działanie po naciśnięciu 'K' (należy uwzględnić to w menu w programie).
4. Napisz program zawierający funkcję, która oblicza sumę liczb podzielnych przez 13 w 30 elementowej tablicy liczb całkowitych. Elementy tablicy powinny być wygenerowane losowo z przedziału -50 do 50 w osobnej funkcji. W programie głównym wyświetlić na monitorze wygenerowaną tablicę 30-elementową (odrębna funkcja) oraz wynik funkcji.
5. Napisz program zawierający funkcję, która znajduje (pierwszą jeżeli jest kilka jednakowych) najmniejszą liczbę dodatnią w pierwszych  $n$  elementach 30 elementowej tablicy liczb całkowitych. Elementy tablicy powinny być wygenerowane losowo z przedziału -50 do 50 w odrębnej funkcji. W programie głównym wyświetlić na monitorze tablicę  $n$ -elementową oraz wynik funkcji.
6. Napisz program zawierający funkcję, która rozpoznaje czy liczba(parametr funkcji) jest pierwsza (zwraca 1 jeżeli jest i 0 jeżeli nie) oraz tworzy tablicę, którą wypełnia 10-cioma kolejnymi liczbami pierwszymi, nie mniejszymi od liczby podanej  $n$  (liczba  $n$  wczytywana z klawiatury w programie głównym). W programie głównym wyświetlić na monitorze utworzoną tablicę 10-elementową.
7. Napisz program zawierający funkcję zamieniającą całkowitą liczbę dziesiętną na liczbę w systemie szesnastkowym. Należy napisać własną implementację i nie korzystać z gotowych funkcji w C.

# Tematy z funkcji:



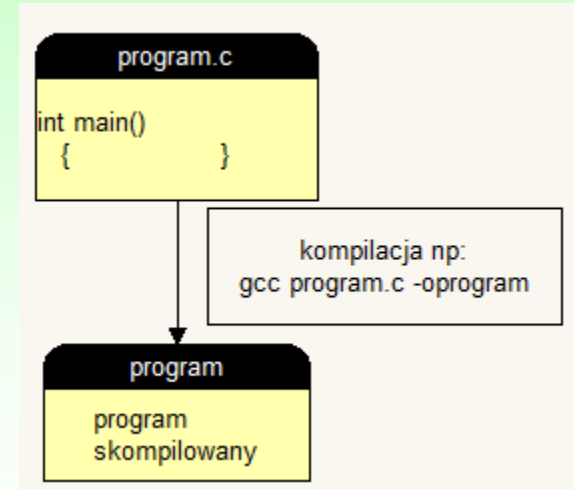
8. Napisz program zawierający funkcję, która oblicza iloczyn liczb dodatnich, nieparzystych w 50 elementowej tablicy liczb całkowitych. Elementy tablicy powinny być wygenerowane losowo w odrębnej funkcji z przedziału od -100 do 100 oraz wyświetlone na monitorze (w innej funkcji) razem z wynikiem zwróconym przez funkcję.

9. Napisać program sortujący (dowolną metodą) tablicę 30 elementową liczb zmiennoprzecinkowych zawierający funkcję zamiany wartości dwóch zmiennych. Tablica powinna być wypełniana w sposób losowy wartościami z przedziału od -100 do 100 z dokładnością do dwóch miejsc po przecinku (odrębna funkcja).

# Parametry programu

```
int main ()  
{  
}
```

```
int main (p,q)  
int p; char *q[ ];  
{  
}
```



Uruchomienie :  
program (ENTER)

Uruchomienie :

q ↓  
„ program „ „ parametr1 „ „ parametr2 „ „ parametr3 „ „ parametr4 „ (ENTER)

p ← 5

Uruchomienie :

q ↓  
„ inny\_prog „ „ par1 „ „ par2 „ (ENTER)

p ← 3

# Typ FILE:

FILE \*fp

fp

`a`	`b`	`x`	`a`	`d`	`a`	`1`	`2`	`2`	`?`	`\n`
`p`	`-`	`+`	`s`	`!`	`\n`					
`.`	`.`	`/`	`6`	`2`	`n`	`q`	`\n`			
`>`	`<`	`\t`	`=`	`=`	`\n`					
`1`	`2`	`3`	`4`	`5`	`6`	`7`	`8`	EOF		

**fp=fopen(*nazwa*, *tryb*)**

***tryb*** – tryb dostępu do pliku:

„r” – czytanie,

„w” – pisanie,

„a” - dopisywanie

**fw=fopen(*nazwa*, *tryb*)**

**Podstawowe działania na zmiennych plikowych:**

**z=getc(fp)**

**putc(z,fp)**

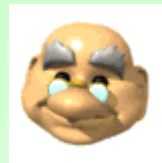
***nazwa*** – nazwa pliku w postaci tekstowej np. „dane.txt” albo odwołanie się do parametrów programu np. q[1]

**Funkcje pozwalające na czytanie i pisanie np. liczb z i do dowolnego pliku:**

**fscanf(fp, „.....”, zm1,zm2,zm3,...)**

**fprintf(fw, „.....”, zm1,zm2,zm3,...)**

# Tematy z plików:



1. Napisz program, który wpisuje do pliku (parametr programu) 40 znaków w wierszach 5-cio znakowych. Przy czytaniu ignoruje znak '\*' i '.' (one nie są wliczane do 40 znaków)

2. Napisz program, który wyświetla na monitorze liczbę wystąpień cyfr i małych liter (każdej osobno) w podanym pliku (parametr programu).

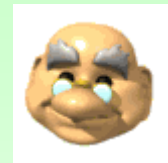
3. Napisz program, który zlicza zapisane linijki w istniejącym pliku (parametr programu) i zapisuje ich liczbę do innego pliku (drugi parametr programu).

4. Napisz program, który kopiuje plik tekstowy (pierwszy parametr programu) do drugiego pliku (drugi parametr programu) w ten sposób, że każdy ciąg spacji redukuje do jednej i na końcu pliku podaje liczbę usuniętych spacji.

5. Napisz program, który kopiuje plik tekstowy (pierwszy parametr programu) do drugiego pliku (drugi parametr programu) w ten sposób, że przy przepisywaniu likwiduje puste linie w pliku (ma usuwać również linie, które zawierają same spacje) i na końcu skopiowanego pliku podaje liczbę usuniętych linii.

6. Napisz program, który kopiuje podany plik (parametr programu) do innego pliku (drugi parametr programu) w wierszach zawierających po 20 znaków i na końcu każdego z tych wierszy dopisuje znak '\*'.(Mamy otrzymać równą kolumnę o szerokości  $20 + '*'$ ).

# Tematy z plików:



7. Napisz program, który kopiując podany plik (parametr programu) do innego pliku (drugi parametr programu), każde zdanie zaczyna od nowej linii i dużej litery. Zakładamy, że zdanie kończy się znakiem kropki '.' i spacji, a na dużą literę zamieniamy tylko małe litery po kropce (inne znaki tj. cyfry, kropki itp. kopiowane są bez zmian).

8. Napisz program, który porównuje pierwsze znaki w kolejnych liniach dwóch plików (parametry programu) i wypisuje na monitorze numery linii, które zaczynały się od tego samego znaku oraz powtarzający się w nich pierwszy znak. (Jeżeli pliki mają różną liczbę wierszy, to "nadmiarowe" wiersze dłuższego pliku nie są analizowane).

9. Napisz program, który przy kopiowaniu jednego pliku do drugiego (parametry programu) zmienia w pierwszej kolumnie małe litery na duże, bez użycia funkcji zamieniającej małe litery na duże dostępnej w bibliotece C (pozostałe znaki kopiuje bez zmian).

10. Napisz program, który znajduje w pliku (parametr programu) cyfry, oblicza ich iloczyn i wypisuje jego wartość na monitorze.

# Tematyka następnych zajęć:

- Sprawdzian z funkcji (napisanie funkcji na zadany temat ze ściśle określonymi parametrami i program główny, w którym napisana funkcja jest uruchamiana).
- Programy z parametrami ( głównej procedury main) wykorzystujące pliki