

## Podstawy programowania, ćw. – lista nr 2.

**Zad. 1.** Napisz łańcuch formatujący dla instrukcji printf wyświetlający na ekranie zmienną rzeczywistą typu double:

- z dokładnością do trzeciego miejsca po przecinku,
- samą część całkowitą,
- w zapisie inżynierskim.

Sekwencja formatująca w instrukcji printf:

%[flagi][szerokość][.precyzja][modyfikatory-rozmiaru-argumentu]znak-typu

flagi: -+# spacja, modyfikator rozmiaru argumentu: FNhLL, znak typu: diouxX eEfgG cs% np

**Zad. 2.** Proszę uzupełnić tabelkę.

Fragment kodu programu	Opis (znaczenie)
<code>const rozmiar_x=5;</code>	.....
.....	Definicja tablicy o rozmiarze <i>rozmiar_x</i> przeznaczonej dla liczb całkowitych.
<code>Tab[2] = - 20 ;</code>	.....
<code>x= .....</code>	Przypisanie zmiennej x wartości umieszczonej w ostatniej komórce tabeli Tab.
<code>scanf( "%d" , &amp;Tab[0] ) ;</code>	.....
.....	To, co wyżej, tylko do przedostatniej i ostatniej komórki w tabeli Tab.
.....	Wyświetlenie na ekranie trzeciej komórki tablicy Tab.
<code>for( i = 0 ; i &lt; rozmiar_x ; i++ ) tablica[i] = 0;</code>	.....
<code>suma=0; for( i = 0 ; i &lt; rozmiar_x ; i++ ) suma = suma + tablica[i];</code>	.....

**Zad. 3.** Określ wartości wyrażeń z uwzględnieniem efektów ubocznych i uzupełnij poniższą tabelę przyjmując następujące definicje

`int j, k=2; float y, x=3.5;`

Wyrażenie	Wynik	Typ wyniku	Efekty uboczne				Uwagi
			j	k	y	x	
<code>y=x</code>	3.5	float			3.5		
<code>j=k=y=x</code>		int					
<code>y=j=x</code>		float					y=j
<code>x*=k+1</code>						10.5	3.5*3
<code>k+=x</code>							k=2+3.5
<code>k &lt;&lt;= (j=x)</code>							k=2<<3
<code>x*=x</code>							x=3.5*3.5
<code>x*=k+(j=y=x+1)</code>							3.5*6
<code>(j=x) &gt; 0</code>							3>0
<code>(y=x) == (j=x)</code>							3.5==3

**Zad. 4.** Narysuj schematy blokowe instrukcji: a) **if**, **if-else**, i pętli: b) **do**, c) **while**, d) **for**.  
e) Uwzględnij także użycie instrukcji **break** i **continue**.

**Zad. 5.** Proszę: a) zrozumieć co robi poniższy kod (wyjaśnić) i narysować schemat blokowy

```

int poz=0;
int max=tab[0];
for( i=1; i<rozmiar_x ; i++ )
    if( max < tab[i] )
    {
        max = tab[i];
        poz = i ;
    }

```

b) uzupełnić funkcję, która wyświetli elementy tablicy i przedstawić algorytm

w postaci listy kroków

```

void WyszwietlTablice( double tablica[ ] )
{
    printf( " Wartości elementów tablicy są równe: \n" );

    for( ..... )
        printf( "Tab[%2d] = %f", i+1 , tablica[i] );
    printf( " Nacisnij dowolny klawisz" );
    getch();
}

```

**Zad. 6.** Proszę uzupełnić tabelkę dla programu z pętlą **while**

<pre> #include &lt;stdio.h&gt; #include &lt;conio.h&gt;  int main(void ) {     int licznik=1;     while (licznik&lt;10) { printf( "-" ); ++licznik; };     getch(); } </pre>	<p>Co będzie wynikiem programu:</p> <p>.....</p> <p>.....</p>
--	---

**Zad. 7. a)** Proszę przeanalizować program i uzupełnić komentarze do programu głównego.

<pre> #include &lt;stdlib.h&gt; #include &lt;stdio.h&gt; #include &lt;conio.h&gt; #include &lt;ctype.h&gt; #include &lt;time.h&gt; #define roz 30  int tab[roz]; int x; int i,j; char opcja;  void sort_int(int *t,int n){     for (i=0;i&lt;n;i++)         for (j=0;j&lt;n;j++) if (*(t+j)&gt;*(t+j+1))             { x=t[j];               t[j]=t[j+1];               t[j+1]=x;             } }  int main(void) { do {     system("cls");     srand(time(NULL));     //.....     for (i=0;i&lt;=roz;i++) tab[i]=rand()%100;     //.....     sort_int(tab,roz);     //wyświetlenie wyniku     for (i=0;i&lt;=roz;i++) printf(" %d",tab[i]);     printf("\nJeszcze raz: ");     opcja=getch(); } while ('T'==toupper(opcja)); } </pre>	<p>b) Proszę narysować schematy blokowe programu i funkcji.</p>
--	---