Podstawy programowania, ćw. – lista nr 4.

Zad. 1. Oznacz zasięgi definicji w podanych poniżej programach. Co wydrukują te programy? Szczególną uwagę proszę zwrócić na kwalifikator globalności: ::identyfikator_globalny.

a)
```
#include <stdio.h>
int R=5;
void main (void)
{ int R=30;
  { int R=100;
    printf("R1=%d ", R);
  }
  printf(R2=%d", R);
}
```

b)
```
#include <stdio.h>
int K=3;
void funkcja (void)
{ printf("K1=%d\n", K); }

void main (void)
{ int K=20;
  printf("K2=%d", K);
  funkcja( );
}
```

c)
```
#include <stdio.h>
int K=3;
int funkcja (void);
{ return (K); }

void main (void)
{ int K=20;
  printf("K3=%d", K+funkcja());
}
```

d)
```
#include <stdio.h>
int R=5;
void main (void)
{ int R=30;
  { int R=100;
    printf("R3=%d", R+::R);
  }
  printf("R4=%d", ::R+R);
}
```

Zad. 2. Jaki będzie wynik działania poniższego programu?

```
#include <stdio.h>
char tab[]="abrakadabra";
char * wsk;
main()
{ wsk=&tab[8];
  putchar(*wsk);              ......
  wsk+=2;
  putchar(*wsk);              ......
  putchar(*(--wsk));          ......
  putchar(tab[1]);            ......
  putchar(*tab);              ......
  wsk=tab+3;
  putchar(*(wsk-1));          ......
  putchar(*wsk);              ......
  putchar('\n'); }
```

Zad. 3. Jaki będzie wyniki działania poniższego programu? Wyjaśnij dlaczego.

```
#include <stdio.h>
void funkc(int * j, int n) //definicja funkcji
{ *j = 1;
  n = *j;
}
main()
{ int i = 2, k = 3;
  funkc(&i, k); //wywołanie funkcji
  printf("i=%d, k=%d", i, k);
}
```

Zad. 4. Przelicz wartości pomiędzy systemami liczbowymi oraz podaj znak kodu ASCII.

Dec	Bin	Oct	Hex	znak
16				
61				
	00110011			3
			A8	
				A
				J
		234		

Zad. 5. Określ wartości wyrażeń i podaj typ wyników. W których wypadkach mamy do czynienia z efektami ubocznymi dotyczącymi zmiennych?

Wyrażenie	Wartość	Typ wyniki
`'J'-('A'-'a')`		
`(7%3<<3)+('B'>'A')`		
`2 | 31&~7^16`		
`12>3?5:4.5/3`		
`k=3,k++,k*=3`		
`5>4>3`		
`((k=4)-(k>1))<<1`		
`6>5?4>3:0.5`		

Zad. 6. a) Zamień pętlę **while** na pętle **do-while** i podaj ile wyniesie **`a`** i **`y`** po wykonaniu pętli:

```
y=1; a=3;
while(y<4)
  {a+=y; y+=1; continue; a=y++;}
```

b) Zamień pętle **for** na pętlę **do-while** i podaj ile wyniesie **`a`** i **`y`** po wykonaniu pętli:

```
for (a=2, y=1; y<5; y+=1)
  {a+=y; continue; a=y++;}
```

c) Dodatkowo narysuj schematy blokowe algorytmów lub rozpisz je w postaci listy kroków.

Zad. 7. Napisz funkcję wyświetlający choinkę według poniższego przykładu. Wysokość choinki powinna być zadawana z programu głównego.

```
   *
  ***
 *****
*******
   #
```

Zad. 8. Proszę odpowiedzieć na pytania:

a) czy poniższa pętla się wykona?

```
for (;;) printf("%d, %c\n",i,i);
```

Odp. .....

b) która komórka tabeli zostanie wydrukowana?

```
i=1; ++i;
printf(" %d", tab[i]);
```

Odp. .....

c) ile razy wykona się pętla (ile wyświetli znaków „x”)?

```
int licznik=1;
while (licznik<10) { printf( "x" ); ++licznik; };
```

Odp. .....

d) podaj przykład instrukcji warunkowej Odp. .....

Zad. 9. Znajdź błędy w instrukcjach (jeśli występują). Jak usunąć błędy?

```
Int a=7;   int b=5.0;   float l2, 22;  !7=<!5;     2||(6=0);    0?1?2;
```