Zadanie 1

Podać wartości, które zostaną wyświetlone na ekranie po wykonaniu poniższego programu. Zachować kolejność.

```
#include <stdio.h>
void f(long  *p1, long  *p2, long x){
   while(p1!=p2){
      if( *p1>x)printf("%ld\n",  *p1);
      ++p1;
   }
}
int main(void){
 long a[15]={5, 9, 6, 1, 7, 4, 8, 2, 1, 3, 7, 6, 5, 1, 9};
 f(a+2, a+11, 5);
 return 0;
}
```

Zadanie 2

Jaką zawartość będzie miała tablica t po wykonaniu programu.

```
#include <stdio.h>
#include <string.h>
void f( char *s){
 while(*s){
  if(!(strlen(s)%2)) *s='!';
  ++s;
  }
}
int main(void){
 char t[20]="programowanie";
 f(t);
 puts(t);
 return 0;
}
```

Zadanie 3

Jakie wartości zostaną wyświetlone po wykonaniu poniższego fragmentu programu:

```
int t[4]={2,5,-4,-3}, *p;
p=t+2; p++;
printf("%d\t %d\t %d\t %d\t %d\t", 2**p,  2**t+4,  2**(t+1),  4*(*p+2),  p-t);
```

Zadanie 4

Podać wartości, które zostaną wyświetlone na ekranie po wykonaniu poniższego programu. Zachować kolejność i format wydruku.

```
#include <stdio.h>
#include <stdlib.h>

void f(float *x, float *y) {
 *y=--*x; x=y; ++*x;
}

int main(void) {
 float *tab,*x;
 tab =(float *) malloc (2*sizeof(float));
 tab[0]=5;
 x=&*(tab+1);
 f(tab,x);
 printf ("%.1f %.2f",tab[0],tab[1]);
 return 0;
}
```

Zadanie 5

Jakie wartości i w jakiej kolejności zostaną wyświetlone na ekranie po wykonaniu poniższego programu. Zachować podział na wiersze.

```
#include <stdio.h>
int f(int *y, int x){
int i, *z=y+2;
for(i=1; *(y+i)>x; i+=2)
  if(*z++>2)
   printf("%d %d \n", i, y[i]);
}
int main(void){
 int tab[8]={5,4,6,10,9,11,2,1};
 f(tab,2);
 return 0;
}
```

Zadanie 6

Zdefiniowano następującą strukturę.

```
struct ulamek
{
int licz;
unsigned mian;
} x,*p;
```

Które z instrukcji nie są poprawne?

```
a) x.licz=3;
b) (&x)->mian=7;
c) *(p.licz)=9;
d) p->mian=1;
```

Zadanie 7

Co się wyświetli wyniku wykonania poniższego programu?

```
#include <stdio.h>
#include <string.h>
int main(){
  char *s1="programowanie ", s2[30]="programowanie proceduralne",
  *s3="program", *w;
  w=s2+strlen(s1);
  puts(w);
  if(!strcmp(s1,s3))
     strcpy(s2,s3);
  puts(s2);
  return 0;
}
```

Zadanie 8

Przy następujących deklaracjach:

```
int *p, **q;
int  (*a)[5];
int m1[2][5];
int m2[2][3];
```

Które z poniższych instrukcji są poprawne?

a. `p=m1[0];`
b. `p=&m2[1][2];`
c. `p=m1;`
d. `a=m1;`
e. `a=m2;`
f. `*q=m2[0];`
g. `q=m2;`

Zadanie 9

Podać wartości, które zostaną wyświetlone na ekranie w wyniku wykonania poniższego programu. Zachować kolejność wydruku.

```
#include <stdio.h>
#define W 3
#define K 4

int sum(int *start , int *stop);

int main (void){
  int tab[W][K]={{2,6,4,5},{0,0,2,6},{2,4,3,7}};
  int *p, *q;
  p = &tab[0][3];
  q = &tab[2][2];

  printf("%d\t", sum(tab[0], p));
  printf("%d\t", sum(p, tab[2]));
  printf("%d\t", sum(&tab[2][0], q));
  printf("%d\t", sum(tab[1]+1, tab[1]+3));
  printf("%d\t", sum(p+3, q+1));

  return 0;
}

int sum(int *start, int *stop){
  int temp=0, *s;
  int i=0;
  for(s=start; s<=stop; s++)
     temp +=*s;
  return temp;
}
```

Zadanie 10

Co się wyświetli wyniku wykonania poniższego programu? Zachowaj kolejność i format wydruku.

```
#include <stdio.h>
int main(void){
 char *w="f-g-e-t-s", *x=w;
 while(*++x);
 do
   if(*--x!='-'){
   putchar(*x);
   putchar('|');
   puts(x);
 }
 while(x-w);
return 0;
}
```

Zadanie 11

W pewnym programie zdefiniowano strukturę:

```
struct my_struct {
    int x;
    int *t1[4];
};
```

Zapisano następujące deklaracje i instrukcje:

```
int t[4][3]={{8,2,3}, {14,9,1}, {11,13,6}, {17,9,5}};
struct my_struct s;

int (*q)[3];

q = t+1;
s.x = (int) sizeof(t)/sizeof(t[0]);
for(int i = 3; i >=0; i--)
    s.t1[i] = t[3-i];
```

Określ, jakie wartości mają poniższe wyrażenia (załóż, że wyrażania są wykonywane sekwencyjnie, tzn. po obliczeniu pierwszego obliczane jest drugie, itd.)?

wyrażenie	wartość
`s.x`	
`*s.t1[2]`	
`s.t1[1][1]`	
`*s.t1[3]`	
`*(*(q+2)+2)`	
`(*(q[-1]+1))++`	
`*(s.t1[3]+1)`	

Zadanie 12

Co zostanie wyświetlone w wyniku wykonania poniższego programu. Zachować kolejność i podział na wiersze.

```
#include <stdio.h>
void write( char *[], int);
int main(void){
 char *countries[5]={"Italy", "Malta", "Spain", "Greece", "Portugal"};
 write(countries,5);
 return 0;
}

void write( char *t[], int n){
int i;
puts(*(t+3));

for(i=0;i<n;++i)
  printf("%c\t", **(t+i));

printf("\n");

for(;n>0;n--)
  puts(t[n-1]);
}
```

Zadanie 13

Wskaż poprawne instrukcje dynamicznego przydziału i zwalniania pamięci dla dwuwymiarowej tablicy t składającej się z N wierszy i M kolumn i typie elementów `int`.

A

```
int **t = (int **) malloc(N * sizeof(int*));
t[0] = (int *) malloc(M * N * sizeof(int));
for(int i = 0; i < M; i++)
     *(t + i) = *t + i * N;

free(*t);
free(t);
```

B

```
int **t = (int **) malloc(N * sizeof(int*));
t[0] = (int *) malloc(M * N * sizeof(int));
for(int i = 0; i < N; i++)
     t[0] = t[0] + i * N;
for(int i = 0; i < N; i++)
     free(t[i]);
free(t);
```

C

```
int **t =(int **) malloc (N * sizeof (int *));
for (int i = 0; i < N ; i ++)
  *(t+i)=(int *) malloc (M * sizeof (int));

free(*t);
free(t);
```

D

```
int **t = (int **) malloc(N * sizeof(int*));
t[0] = (int *) malloc(N * M * sizeof(int));
for(int i = 0; i < N; i++)
     t[i] = *t + i * M;
free(*t);
free(t);
```

Zadanie 14

Podać wartości, które zostaną wyświetlone w wyniku wykonania poniższego programu. Zachować kolejność, podział na wiersze i format wydruku.

```
#include <stdio.h>
#define N 3
void f(double (*x)[N],int y){
 int i;
 for (i=0;i<y;i++)
   printf("%.2f ",**(x+i));
 double (*p)[N]=x+1;
 printf("\n%.2f  %.2f  %.2f",  *p[0], *(*p+2), *p[-1]);
}
int main(void){
 double t[5][3]={{4.7,3.5},{2.2,15.5},{-3.0,4.0,5.0},{-5.5,9.1,5.5}};
 f(t,5);
 return 0;
}
```

Zadanie 15

Dla każdej z podanych poniżej deklaracji funkcji wskazać, które instrukcje zawierające wywołanie tej funkcji są poprawne (zostaną zaakceptowane przez kompilator zgodny ze standardem języka C). Przyjąć, że wywołanie funkcji zostało poprzedzone następującymi deklaracjami zmiennych:

```
int m, n;
double x, y;
int *p;
int t[10];
struct towar s;
```

Deklaracje funkcji:

```
(1) int f(int);
(2) int f(long);
(3) double f(double);
(4) long f(void);
(5) void f(void);
(6) int f(int*);
(7) int f(double*);
(8) int f(struct towar);
(9) struct towar f(int);
(10) void f(int, int);
(11) int f(int*, int);
```

Instrukcje zawierające wywołanie funkcji:

```
a) f(2);
b) f(3.0);
c) f(t);
d) x=f(t);
e) x=f(2)+3;
f) m=f(y+3);
g) f();
h) y=f();
i) f(&n);
j) f(&y);
k) f(t, m);
l) f(m, &n);
m) f(&x, n);
n) y=f(s)+5;
o) s=f(8);
```

Zadanie 16

Podać wartości, które zostaną wyświetlone na ekranie w wyniku wykonania poniższego programu. Zachować kolejność i podział na wiersze.

```
struct wektor{
  int y, x;
};
void f(struct wektor w){
 printf("%d %d\n", w.y, w.x);
}
int main(void){
 struct wektor a[5]={{17,2},{3,9},{8,5},{6,10},{7,0}};
 int i=0,k =sizeof a/sizeof a[0];
 for(; i<k; ++i) if(a[i].x < a[i].y) f(a[i]);
 return 0;
}
```

Zadanie 17

Struktura poniżej opisuje punkt na płaszczyźnie. Wybierz brakującą instrukcję z listy propozycji, aby sprawdzić w której ćwiartce układu współrzędnych leży punkt.

```
struct punkt
{
float x;
float y;
};
struct punkt  tp[10];
int kwadrat, i;

for(i=0;i<10;i++)
 ***************
```

a)

```
if (tp[i].x<0||tp[i].y<0) kwadrat=3;
else if (tp[i].x>0||tp[i].y>0)kwadrat=1;
else if (tp[i].x<0||tp[i].y>0)kwadrat=2;
else kwadrat=4;
```

b)

```
if (tp[i].x * tp[i].y<0)kwadrat= tp[i].x<0?2:4;
else if (tp[i].x*tp[i].y>0) kwadrat = tp[i].x>0?1:3;
```

c)

```
if (tp[i].x * tp[i].y<0)kwadrat= tp[i].x<0?1:3;
else if (tp[i].x*tp[i].y>0) kwadrat = tp[i].x>0?2:4;
```

d)

```
żadna z powyższych
```

Zadanie 18

Podać wartości, które zostaną wyświetlone na ekranie w wyniku wykonania programu. Zachować kolejność i podział na wiersze.

```
#include <stdio.h>

struct abc{
   int x;
   int y;
};

int f(struct abc a, struct abc *b)
{
 printf("%d %d\n", a.y, b->x);
 a.x = b->y;
 a.y = a.x - a.y;
 b->x = a.y;
 ++b->y;
 return b->y - (*b).y;
}

int main(void)
{
 struct abc u ={2, 3}, v ={1, 8};
 int k = f(u, &v);
 printf("%d %d %d %d %d\n", u.x, u.y, v.x, v.y, k);
 return 0;
}
```