

## Imię i nazwisko

Wypełnij drukowanymi literami

## Numer indeksu

czas pisania: 75 minut, 17 luty 2016

Uwaga: w przypadku wszystkich programów należy założyć, że dołączone są biblioteki `iostream`, `stdlib.h` oraz dostępna jest przestrzeń nazw `std`. Sprawdzaniu podlegają jedynie miejsca wyznaczone na odpowiedź. W przypadku stwierdzenia błędu lub niejednoznaczności w pytaniu, należy czytelnie napisać odpowiedni komentarz wyjaśniający napotkany problem. Test oceniany jest w skali 0-100 pkt (próg zaliczenia = 50%).

### Zad. 1. (20 pkt. = 5\*4)

Wyróżnione pola uzupełnij, tak aby program poprawnie się kompilował, nie powodował błędów wykonania oraz wypisał na wyjście:

**15913**

Należy także zadbać o jawne rzutowanie we wszystkich miejscach, w których wystąpi niezgodność typów.

Podając odpowiedź:

- należy wpisać BŁĄD jeśli rozwiązanie nie istnieje,
- nie należy używać ; oraz ,
- Rozmiar alokowanych tablic powinien być minimalny.

```
int main() {
    int **tab, i;

    tab = (int **) malloc( 4*sizeof(int*) );
    for ( i=0; i < 4; i++ ) {

        tab[i] = (int *) malloc( 2*sizeof(int) );
        tab[i][0] = 2*i;

        tab[i][1] = 2*i+1;
    }
    for ( i=0; i < 4; i++ )
        cout << tab[i][0] + tab[i][1];
    return 0;
}
```

### Zad. 2. (20 pkt. = 4\*5)

Podaj wartości zwrócone przez podane poniżej wywołania funkcji.

#### Odpowiedź:

f2( 3 )    **1**

f2( 7 )    **1**

f2( 11 )   **1**

f2( 15 )   **1**

```
int f2( int );

int f1( int a, int b ) {
    return ( a < b ) ? f2( b-a ) : f2( a-b );
}

int f2( int a ) {
    if ( a<=1 )
        return 1;
    else
        return f1( a, (2*a+3)%a+1 );
}
```

### Zad. 3. (20 pkt. = 4\*5)

Wyróżnione pola uzupełnij, tak aby program poprawnie się kompilował i nie powodował błędów wykonania. Podając odpowiedź:

1. należy wpisać BŁĄD jeśli rozwiązanie dla danego pola nie istnieje,
2. należy używać znaków: \*.<>-[]& oraz cyfr i liter.

```
typedef struct A {
    struct A *n;
    struct {
        int x;
        struct {
            int y;
        } z;
    } t[3];
} tA;

int main() {
    tA *s1, s2[4], s3;

    s1 = &(s2[3]);
    s1->n = &s3;

    s1->n->t[0].x = s2[1].n->t[1].z.y;

    *s2 = *((*(s1-1)).n);
    return 0;
}
```

**Zad. 4.** (18 pkt. = 6\*3)

Obok każdej instrukcji "cout" podaj tekst, który zostanie wypisany na wyjściu w wyniku jej wykonania. Podając odpowiedź należy wpisać BŁĄD jeśli nie można jednoznacznie stwierdzić co zostanie wypisane na ekran.

```
int a;
typedef struct { int x, y; } pT;
pT h[3];
void f( pT c, pT *a, pT **b ) {
    a->y = 4;
    c.x = 7;
    b[0]->x = b[0]->y + 3;
}
int main() {
    int *r, w[3] = { 0, 3, 6 };
    pT *y = h+1;

    r = w+1;
    {
        int a;
        3      cout << sizeof( w ) / sizeof( a );

        BŁĄD      cout << a / *r ;
    }
    31      cout << ((12 ^ 3) | 24);
            f( h[0], h+1, &y );
    0      cout << h[0].x;

    4      cout << h[1].y;

    0      cout << h[2].x;
            return 0;
    }
```

**Zad. 5.** (22 pkt.=5\*5,5) Podaj co pojawi się na wyjściu w wyniku kolejnych wywołań procedury printt (w polu oznaczonym jako "wykonanie x" wypisz znaki, które pojawią się na wyjściu w wyniku realizacji instrukcji wykonanej podczas wykonania numer x tej procedury).

**Odpowiedź:**

Wykonanie 1:

2 3 4 7 0 5 0 7 4 3

Wykonanie 2:

2 3 4 7 0 5 2 0 4 3

Wykonanie 3:

2 3 4 2 4 5 2 0 4 3

Wykonanie 4:

2 2 4 2 4 5 2 0 4 3

```
#define SIZE 10
int test( int *t, int k, int n ) {
    while ( k < n-- ) {
        k += t[k];
        t[n] = t[k%n];
    }
    return n;
}

void printt( int *t ) {
    int i;
    for ( i=0; i < SIZE; i++ )
        cout << t[i] << " ";
    cout << endl;
}

int t[SIZE], n=SIZE-1, k=0, i;
int main( ) {
    for ( ; i < SIZE; i++ )
        t[i] = ((i*i + 5) % 20)/2;
    printt( t );
    while ( n ) {
        n = test( t, k, n );
        printt( t );
    }
    return 0;
}
```