

**Imię i nazwisko**

Wypełnij drukowanymi literami

**Numer indeksu**

czas pisania: 90 minut, 3 luty 2014

Uwaga: w przypadku wszystkich programów należy założyć, że dołączone są biblioteki `iostream` i `stdlib.h` oraz dostępna jest przestrzeń nazw `std`.

**Zad. 1.** (12 pkt.) Przy wybranych liniach kodu znajdują się komentarze określające ich numery. Podaj poniżej numery linii, które powodują błąd kompilacji programu:

Odpowiedź: **2,5,10,12**

Uwaga: kompilator C++ uzna linię nr 5 za błąd. Kompilator C da jedynie ostrzeżenie. Z tego względu przyjmujemy  $x=3$ , z maksymalną liczbą punktów za zadanie równą 12.

Sposób liczenia punktów ( $x$  jest równe liczbie linii powodujących błąd kompilacji):

- za każdą poprawnie wybraną linię:  $12/x$  punktów,
- za każdą niepoprawnie wybraną linię:  $-12/x$  punktów,
- jeśli liczba niepoprawnie wybranych instrukcji przekracza liczbę poprawnie wybranych instrukcji, to liczba punktów za zadanie wynosi 0.

```
void f( const int *w ) {
    char *s; //1
    w[0] = 1; //2
    w++; //3
    w = NULL; //4
    s = w; //5
    s[0] = 0; //6
}
void g( int t[2][2] ) {
    t[0][0] = t[1][1]; //7
}
int main() {
    int t[10], u[10][2]; //8
    f( t ); //9
    g( t ); //10
    g( u ); //11
    t++; //12
    return 0; }
```

**Zad. 2.** (14 pkt. =  $2*7$ )

Wyróżnione pola uzupełnij, tak aby program poprawnie się kompilował oraz wypisywał na ekran:

**234.** Podając odpowiedź:

- nie należy używać znaków: `;`, `()` oraz cyfr
- należy wpisać BŁĄD jeśli rozwiązanie nie istnieje
- należy wpisać BRAK jeśli pole powinno pozostać puste.

```
int f( void ) {
    static int x;

    cout << ++x + 1;
    return x;
}
int main() {
    f() + f() + f();
    return 0;
}
```

**Zad. 3.** (16 pkt. =  $4*4$ )

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

```
struct A { int a; } ts[4];

void pp1( struct A *ts ) {
    ts++;
    (*ts).a = 1;
}
void pp2( struct A ts ) {
    ts.a += 2;
}
void pp3( struct A ts[] ) { ts[0].a = 3; }
void pp4( struct A ts[2] ) { ts[1].a = 4; }
int main() {
    pp1( ts );
    pp1( ts+2 );
    pp2( ts[0] );
    pp2( *(ts+1) );
    pp3( ts );
    pp4( ts );
    cout << ts[0].a << endl;
    cout << ts[1].a << endl;
    cout << ts[2].a << endl;
    cout << ts[3].a << endl;    return 0;}

3
4
0
1
```

**Zad. 4.** (16 pkt.)

Poniżej podaj tekst, który zostanie wypisany na ekran w wyniku uruchomienia podanego programu. Wpisz BŁĄD w przypadku błędu kompilacji, błędu wykonania programu, lub w przypadku gdy nie można jednoznacznie określić tekstu, który zostanie wypisany na ekran.

Odpowiedź:

**ee-i**

```
char * napisy( char *s, int n, char *t ) {
    int i = 0, j = 0;
    while ( n >= ++i ) {
        if ( *(s+i) == s[n-1] ) {
            continue;
        }
        t[j++] = s[n-2];
        n -= (i % n)/4;
    }
    t[4] = '\0';
    return t;
}
int main() {
    char x[] = "0-d-p-o-w-i-e-d-z-P-r-e-c-y-z-y-j-n-i-e";
    char w[20];
    cout << napisy( x+2, 12, w ) << endl;
    return 0;
}
```

**Zad. 5.** (16 pkt. = 4\*4)

Jaka wartość zostanie zwrócona przez poniższe wywołania funkcji:

rekurencja2( 1 ) **2**rekurencja2( 2 ) **3**rekurencja2( 3 ) **6**rekurencja2( 4 ) **9**

```
int rekurencja2( int n );
int rekurencja1( int n, int k ) {
    return rekurencja2( (n-1) % k )
        + rekurencja2( k % n );
}
int rekurencja2( int n ) {
    if ( n )
        return rekurencja1( n, n + 2 );
    else
        return 1;
}
```

**Zad. 6.** (26 pkt. = 13\*2) Obok każdej instrukcji "cout", podaj tekst, który zostanie wypisany. Wpisz BŁĄD jeśli nie można tego jednoznacznie określić lub gdy instrukcja powoduje błąd wykonania programu.

```
#define P d2+1
int main() {
    int *p = (int *) malloc( 10*sizeof(int) );
    int t[10], d1 = 5, d2 = 5;
    char s[] = {"Napis"}, c1 = 100, c2 = '1';

    BŁĄD cout << (int)sizeof(p)-(int)sizeof(t) << endl;
    200 cout << c1 * 2 << endl;
    BŁĄD cout << c1 + (c1++) << endl;
    15 cout << (13 & 9) + (3 << 1) << endl;
    14 cout << (13 | 9) + (3 >> 1) << endl;
    4 cout << (13 ^ 9) << endl;
    t[0] = (1 << 4) | (1 << 2);
    20 cout << *t << endl;
    if ( t == NULL && c2++ )
        d2 = 0;
    5+5 cout << d1 << '+' << d2 << endl;
    6 cout << sizeof( s ) << endl;
    p = &(t[5]);
    5 cout << p - t << endl;
    2 cout << (char) (c2 + 1) << endl;
    3.5 cout << 15 / 2 / 2.0 << endl;
    6 cout << P % 3 << endl; return 0;}
```

**Imię i nazwisko**

Wypełnij drukowanymi literami

**Numer indeksu**

czas pisania: 90 minut, 3 luty 2014

Uwaga: w przypadku wszystkich programów należy założyć, że dołączone są biblioteki `iostream` i `stdlib.h` oraz dostępna jest przestrzeń nazw `std`.

**Zad. 1.** (12 pkt.) Przy wybranych liniach kodu znajdują się komentarze określające ich numery. Podaj poniżej numery linii, które powodują błąd kompilacji programu:

Odpowiedź: **3,4,9,11**

Uwaga: kompilator C++ uzna linię nr 4 za błąd. Kompilator C da jedynie ostrzeżenie. Z tego względu przyjmujemy  $x=3$ , z maksymalną liczbą punktów za zadanie równą 12.

Sposób liczenia punktów ( $x$  jest równe liczbie linii powodujących błąd kompilacji):

- za każdą poprawnie wybraną linię:  $12/x$  punktów,
- za każdą niepoprawnie wybraną linię:  $-12/x$  punktów,
- jeśli liczba niepoprawnie wybranych instrukcji przekracza liczbę poprawnie wybranych instrukcji, to liczba punktów za zadanie wynosi 0.

```
void f( const int *w ) {
    char *s; //1
    w++; //2
    w[0] = 1; //3
    s = w; //4
    s[0] = 0; //5
    w = NULL; //6
}
void g( int t[2][2] ) {
    t[0][0] = t[1][1]; //7
}
int main() {
    int t[10], u[10][2]; //8
    t++; //9
    f( t ); //10
    g( t ); //11
    g( u ); //12
    return 0; }
```

**Zad. 2.** (14 pkt. =  $2*7$ )

Wyróżnione pola uzupełnij, tak aby program poprawnie się kompilował oraz wypisywał na ekran:

**123.** Podając odpowiedź:

- nie należy używać znaków: `;`, `()` oraz cyfr
- należy wpisać BŁĄD jeśli rozwiązanie nie istnieje
- należy wpisać BRAK jeśli pole powinno pozostać puste.

```
int f( void ) {
    static int x;

    cout << x++ + 1;
    return x;
}
int main() {
    f() + f() + f();
    return 0;
}
```

**Zad. 3.** (16 pkt. =  $4*4$ )

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

```
struct A { int a; } ts[4];

void pp1( struct A *ts ) {
    ts++;
    (*ts).a = 2;
}
void pp2( struct A ts ) {
    ts.a += -1;
}
void pp3( struct A ts[] ) { ts[0].a = 4; }
void pp4( struct A ts[2] ) { ts[1].a = 3; }
int main() {
    pp1( ts );
    pp1( ts+2 );
    pp2( ts[0] );
    pp2( *(ts+1) );
    pp3( ts );
    pp4( ts );
    cout << ts[0].a << endl;
    cout << ts[1].a << endl;
    cout << ts[2].a << endl;
    cout << ts[3].a << endl;    return 0;}

4
3
0
2
```

**Zad. 4.** (16 pkt.)

Poniżej podaj tekst, który zostanie wypisany na ekran w wyniku uruchomienia podanego programu. Wpisz BŁĄD w przypadku błędu kompilacji, błędu wykonania programu, lub w przypadku gdy nie można jednoznacznie określić tekstu, który zostanie wypisany na ekran.

Odpowiedź:

**dd-e**

```
char * napisy( char *s, int n, char *t ) {
    int i = 0, j = 0;
    while ( n >= ++i ) {
        if ( *(s+i) == s[n-1] ) {
            continue;
        }
        t[j++] = s[n];
        n -= (i % n)/4;
    }
    t[4] = '\0';
    return t;
}
int main() {
    char x[] = "0-d-p-o-w-i-e-d-z-P-r-e-c-y-z-y-j-n-i-e";
    char w[20];
    cout << napisy( x+2, 12, w ) << endl;
    return 0;
}
```

**Zad. 5.** (16 pkt. = 4\*4)

Jaka wartość zostanie zwrócona przez poniższe wywołania funkcji:

rekurencja2( 1 ) **4**rekurencja2( 2 ) **6**rekurencja2( 3 ) **12**rekurencja2( 4 ) **18**

```
int rekurencja2( int n );
int rekurencja1( int n, int k ) {
    return rekurencja2( (n-1) % k )
        + rekurencja2( k % n );
}
int rekurencja2( int n ) {
    if ( n )
        return rekurencja1( n, n + 2 );
    else
        return 2;
}
```

**Zad. 6.** (26 pkt. = 13\*2) Obok każdej instrukcji "cout", podaj tekst, który zostanie wypisany. Wpisz BŁĄD jeśli nie można tego jednoznacznie określić lub gdy instrukcja powoduje błąd wykonania programu.

```
#define P d2+2
int main() {
    int *p = (int *) malloc( 10*sizeof(int) );
    int t[10], d1 = 4, d2 = 4;
    char s[] = {"Napisy"}, c1 = 99, c2 = '2';

    BŁĄD cout << (int)sizeof(p)-(int)sizeof(t) << endl;
    198 cout << c1 * 2 << endl;
    14 cout << (12 & 9) + (3 << 1) << endl;
    BŁĄD cout << c1 + (c1++) << endl;
    14 cout << (12 | 9) + (3 >> 1) << endl;
    5 cout << (12 ^ 9) << endl;
    t[0] = (1 << 4) | (2 << 1);
    20 cout << *t << endl;
    if ( t == NULL && c2++ )
        d2 = 0;
    4+4 cout << d1 << '+' << d2 << endl;
    7 cout << sizeof( s ) << endl;
    p = &(t[5]);
    5 cout << p - t << endl;
    3 cout << (char) (c2 + 1) << endl;
    3.5 cout << 15 / 2 / 2.0 << endl;
    6 cout << P % 3 << endl;    return 0;}
```