

Imię i nazwisko

Wypełnij drukowanymi literami

Numer indeksu

czas pisania: 90 minut, 17 lutego 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:

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 d1( int *w ) {  
    w += 100;          // 1  
}  
int d2( int t ) {  
    return ++t;        // 2  
}  
int main() {  
    int t[5];           // 3  
    d1 ( *t );          // 4  
    d2( t[3] );         // 5  
    d1( d2 ( 5 ) );     // 6  
    return '0';        // 7  
}
```

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

Wyróżnione pola uzupełnij, tak aby program poprawnie się kompilował oraz wypisywał na ekran: **09**. Podając odpowiedź:

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

```
int x;  
int main() {  
    cout << _ _ _ _ _ _ _ _ ;  
    if ( x ) {  
        char y = _ _ _ _ _ _ _ _ ;  
        cout << y;  
    }  
    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[3]; } str[3];  
void pp1( int *x ) {  
    x++;  
    *x = 1; }  
void pp2( int x ) {  
    x++;  
    x = 2; }  
void pp3( struct A *str ) {  
    str[0].a[0] = 3;  
}  
void pp4( struct A str[] ) {  
    str[0].a[1] = 4;  
    pp3( str+1 );  
}  
void pp5( struct A str ) {  
    str.a[2] = 5;  
    pp3( &str );  
}  
int main() {  
    pp1( str[2].a );  
    pp2( str[1].a[1] );  
    pp3( str+1 );  
    pp4( &(str[1]) );  
    pp5( str[2] );  
    cout << str[1].a[0] << endl;  
    cout << str[1].a[1] << endl;  
    cout << str[0].a[0] << endl;  
    cout << str[2].a[1] << endl;  
    return 0; }
```

_ _ _ _

_ _ _ _

_ _ _ _

_ _ _ _

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:

```
void f( char *s, int n, int k ) {
    if ( k ) {
        char t = s[k];
        s[k] = s[n-k];
        s[n-k] = t;
        f( s, n, --k );
    }
}

int main() {
    char x[] = "Halabala";
    f( x, 7, 5 );
    cout << x << endl;
    return 0;
}
```

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

Jaka wartość zostanie
zwrócona przez poniższe
wywołania funkcji:

g(2) _ _ _ _ _

g(3) _ _ _ _ _

$g(4) = \underline{\hspace{1cm}} \underline{\hspace{1cm}} \underline{\hspace{1cm}} \underline{\hspace{1cm}} \underline{\hspace{1cm}} \underline{\hspace{1cm}}$

g(6) _ _ _ _ _

```
int g( int n ) {
    cout << "g(" << n << ") ";
    if ( n / 3 == 0 )
        return 2;
    else
        return g( n - 1 - (n%3) ) + g( n/2 );
}
```

Zad. 6. (24 pkt. = 12*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.

Uwaga: zakładamy,
że wywołanie
funkcji malloc
powiodło się.

```
#define W u+h
int main() {
    int *p = (int *) malloc( 100*sizeof(int) );
    char c = '4';
    int h=3, u=6, *v = &(p[20]), y[] = {1,2,3,4};

    cout << (8 | 3) << endl;
    cout << (19 & 9) << endl;
    cout << (9 ^ 2) << endl;
    v += (h << 2);
    cout << v << endl;
    cout << ( (c+1) - (int)'0' ) << endl;
    cout << ( (v-p)/sizeof(int) ) << endl;
    cout << ( W*W ) << endl;
    cout << sizeof( p ) << endl;
    cout << ( (*p) ? h : u ) << endl;
    cout << 10.0 / 2 / 2 << endl;
    cout << *y+3 << endl;
    cout << &(y[3]) - &(*y) << endl;
    return 0;
}
```

Imię i nazwisko

Wypełnij drukowanymi literami

Numer indeksu

czas pisania: 90 minut, 17 lutego 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:

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 d1( int *w ) {  
    w += 100;      // 1  
}  
int d2( int t ) {  
    return ++t;    // 2  
}  
int main() {  
    int t[5];      // 3  
    d2( t[3] );    // 4  
    d1 ( *t );     // 5  
    d1( d2 ( 5 ) ); // 6  
    return '0';    // 7  
}
```

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

Wyróżnione pola uzupełnij, tak aby program poprawnie się kompilował oraz wypisywał na ekran: **16**. Podając odpowiedź:

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

```
int x;  
int main() {  
    cout << _ _ _ _ _ _ _ _ ;  
    if ( x ) {  
        char y = _ _ _ _ _ _ _ _ ;  
        cout << (char)(y-1);  
    }  
    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[3]; } str[3];  
void pp1( int *x ) {  
    x++;  
    *x = 6; }  
void pp2( int x ) {  
    x++;  
    x = 3; }  
void pp3( struct A *str ) {  
    str[0].a[0] = 4;  
}  
void pp4( struct A str[] ) {  
    str[0].a[1] = 5;  
    pp3( str+1 );  
}  
void pp5( struct A str ) {  
    str.a[2] = 1;  
    pp3( &str );  
}  
int main() {  
    pp1( str[2].a );  
    pp2( str[1].a[1] );  
    pp3( str+1 );  
    pp4( &(str[1]) );  
    pp5( str[2] );  
    cout << str[1].a[0] << endl;  
    cout << str[1].a[1] << endl;  
    cout << str[0].a[0] << endl;  
    cout << str[2].a[1] << endl;  
    return 0; }
```

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:

```
void f( char *s, int n, int k ) {
    if ( k ) {
        char t = s[k-1];
        s[k] = s[n-k-1];
        s[n-k-1] = t;
        f( s, n, --k );
    }
}

int main() {
    char x[] = "Halabala";
    f( x, 7, 5 );
    cout << x << endl;
    return 0;
}
```

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

Jaka wartość zostanie
zwrócona przez poniższe
wywołania funkcji:

g(2) _ _ _ _ _

g(3) _ _ _ _ _

g(4) _ _ _ _ _

g(6) _ _ _ _ _

```
int g( int n ) {
    cout << "g(" << n << ") ";
    if ( n / 3 == 0 )
        return 3;
    else
        return g( n - 1 - (n%3) ) + g( n/2 );
}
```

Zad. 6. (24 pkt. = 12*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.

Uwaga: zakładamy,
że wywołanie
funkcji malloc
powiodło się.

```
#define W u+h
int main() {
    int *p = (int *) malloc( 100*sizeof(int) );
    char c = '4';
    int h=5, u=4, *v=&(p[30]), y[]={4,3,2,1,0};

    cout << 10.0 / 2 / 2 << endl;
    cout << (12 | 3) << endl;
    cout << (19 & 7) << endl;
    cout << (4 ^ 3) << endl;
    v += (h << 2);
    cout << v << endl;
    cout << ( (c+2) - (int)'0' ) << endl;
    cout << ( (v-p)/sizeof(int) ) << endl;
    cout << ( W*W ) << endl;
    cout << sizeof( p ) << endl;
    cout << ( (*p+1) ? h : u ) << endl;
    cout << *y+3 << endl;
    cout << &(y[3]) - &(*y) << endl;
    return 0; }
```