
Imię i nazwisko

Wypełnij drukowanymi literami

Numer indeksu

czas pisania: 75 minut, 16 luty 2017

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. = 2*10 pkt.) Wyróżnione pola uzupełnij, tak aby program poprawnie się kompilował oraz wypisywał na wyjście:

345

Podając odpowiedź:

- 1) nie należy używać znaków: `;`, `=()`
- 2) należy wpisać BŁĄD jeśli rozwiązanie nie istnieje.

```
int f( void ) {  
  
    static int x;  
  
    cout << ++x + 2;  
    return x;  
}  
int main() {  
    f();  
    f();  
    f();  
    return 0;  
}
```

Zad 2. (20 pkt. = 5*4 pkt.) Wyróżnione pola uzupełnij, tak aby program poprawnie się kompilował oraz wypisywał na wyjście:

60

Podając odpowiedź:

- 1) jeśli w danym polu zostanie podane wyrażenie arytmetyczne, to powinno ono w trakcie wykonania programu ewaluować się do najmniejszej wartości gwarantującej poprawne wykonanie programu
- 2) należy wpisać BŁĄD jeśli rozwiązanie nie istnieje.

```
int main() {  
  
    int ** tab, i, suma = 0 ;  
  
    tab = (int **) malloc( 20*sizeof(int*) ) ;  
    for ( i=0; i < 20; i++ ) {  
  
        tab[i] = (int *) malloc( 2*sizeof(int) ) ;  
        tab[i][0] = 1;  
        tab[i][1] = 2;  
    }  
    for ( i=0; i <= 19 ; i++ )  
        suma += tab[i][0] + tab[i][1];  
    cout << suma;  
    return 0;  
}
```

Zad. 3. (19 pkt.) Podaj co pojawi się na wyjściu w wyniku wykonania podanego programu.

Odpowiedź: **PDrgiizy**

```
void dziwne_kopiowanie( char *s, char *d ) {  
    while ( *d )  
        *(++s) = *(d++);  
}  
  
int main() {  
    char s1[100] = "Pierwszy";  
    char s2[] = "Drugi";  
    dziwne_kopiowanie( s1, s2 );  
    dziwne_kopiowanie( s1+2, &s2[3] );  
    cout << s1;  
    return 0;  
}
```

Zad. 4. (21 pkt. = 7*3 pkt.) Podaj tekst, który zostanie wypisany na wyjściu w wyniku wykonania poszczególnych instrukcji "cout". Podając odpowiedź należy wpisać BŁĄD jeśli nie można jednoznacznie stwierdzić co zostanie wypisane na ekran. Przy miejscu na odpowiedź oznaczonym etykietą "Instrukcja x" spisz tekst wypisany przez instrukcję "cout" przy której widnieje komentarz "Instrukcja x". Kodowanie liczb w systemie binarnym przyjmujemy tak jak omówiono na wykładzie, tzn. U2.

Odpowiedzi:

Instrukcja 1: **2.5**

Instrukcja 2: **7**

Instrukcja 3: **BŁĄD**

Instrukcja 4: **BŁĄD**

Instrukcja 5: **-22**

Instrukcja 6: **8**

Instrukcja 7: **3**

```
#define W u-h
int p[100];
int main() {
    int h=8, u=2, *v = &(p[30]), y[] = {5,3,6,1,8};

    cout << 10.0 / 2 / 2; // Instrukcja 1

    cout << (4 ^ 3); // Instrukcja 2

    v += (h << 2);
    cout << v; // Instrukcja 3

    cout << ( (v-p)/sizeof(int) ); // Instrukcja 4

    cout << ( W*W ); // Instrukcja 5

    cout << *y+3; // Instrukcja 6

    cout << &(y[3]) - &>(*y); // Instrukcja 7

    return 0;
}
```

Zad. 5. (20 pkt. = 4*5 pkt.) Podaj tekst, który zostanie wypisany na wyjściu w wyniku wykonania poszczególnych instrukcji "cout". Podając odpowiedź należy wpisać BŁĄD jeśli nie można jednoznacznie stwierdzić co zostanie wypisane na ekran. Przy miejscu na odpowiedź oznaczonym etykietą "Instrukcja x" spisz tekst wypisany przez instrukcję "cout" przy której widnieje komentarz "Instrukcja x".

Odpowiedzi:

Instrukcja 1: **7**

Instrukcja 2: **0**

Instrukcja 3: **1**

Instrukcja 4: **3**

```
struct A { int a[3]; } str[3];
void pp1( int *x ) {
    x++;
    *x = 3;
}
void pp2( int x ) {
    x++;
    x = 5;
}
void pp3( struct A *str ) {
    str[0].a[0] = 7;
}
void pp4( struct A str[] ) {
    str[0].a[1] = 1;
    pp3( str+1 );
}
void pp5( struct A str ) {
    str.a[2] = 1;
    pp3( &str );
}
int main() {
    pp1( str[2].a );
    pp3( str+1 );
    pp4( &(str[1]) );
    pp2( str[1].a[1] );
    pp5( str[2] );
    cout << str[1].a[0]; // Instrukcja 1
    cout << str[0].a[0]; // Instrukcja 2
    cout << str[1].a[1]; // Instrukcja 3
    cout << str[2].a[1]; // Instrukcja 4
    return 0;
}
```