

Imię i nazwisko

Wypełnij drukowanymi literami

Numer indeksu

czas pisania: 75 minut, 23 styczeń 2015

Uwaga: w przypadku wszystkich programów należy założyć, że dołączone są biblioteki `iostream` i `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.

Zad. 1. (18 pkt. = 3*6) Podaj wartości wybranych elementów tablicy `liczby` tuż po zakończeniu realizacji funkcji `pytanie1` w podanym programie.

Odpowiedź:`liczby[0] = _____``liczby[1] = _____``liczby[2] = _____`

```
int liczby[] = { 7, 9, 12, 6, 4, 9 };
int pytanie1( int t[], int l, int r ) {
    if ( l < r ) {
        t[l] = t[r] - pytanie1(t+l/2, l+1, r/2) % 3;
    }
    return t[(l+r)/2];
}
int main() {
    pytanie1(liczby, 0, 5);
    return 0;
}
```

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

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

- nie należy używać znaków: `=;`, `()[]` oraz dużych liter alfabetu
- należy wpisać **BŁĄD** jeśli rozwiązanie nie istnieje
- należy wpisać **BRAK** jeśli pole powinno pozostać puste.

```
void pytanie2( _____ napis ) {
    int a, b;
    for ( a=0; napis[a] != '\0'; a++ );

    for ( b=0; b < a/2; b++ ) {
        int c = napis[_____], x = _____;
        napis[b] = napis[x];
        napis[x] = c;
    }
}
int main() {
    char napis[] = { "WETIPG" };

    pytanie2( _____ );
    cout << napis;
    return 0;
}
```

Zad. 3. (25 pkt. = 5*5)

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.

```
void f( char *t ) { t[1] = 'D'; }
void g( char t ) { t = 'E'; }
```

```
int main() {
    int a[] = { 1, 2, 3 };
    int *ptr = &(a[2]);
    char text[] = { "abc" };
```

```
    cout << (int) (9/2/2.0);
```

```
    g( text[2] );
    cout << text;
```

```
    f( text + 2 );
    cout << sizeof( text );
```

```
    cout << (--ptr) - a;
```

```
    cout << text;
    return 0;
```

```
}
```

Zad. 4. (17 pkt. = 5+6+6)
Jaka wartość zostanie
zwrócona przez poniższe
wywołania funkcji:

pytanie4(5,0) _____

pytanie4(5,-1) _____

pytanie4(5,2) _____

```
int sgn( int x ) {  
    return ((x > 0) ? x : -x);  
}  
  
int pytanie4( int a, int b ) {  
    if ( a <= b || b == 0 )  
        return b-a;  
    else  
        return pytanie4( b, pytanie4( a, sgn(a-2*b)%sgn(b) ) );  
}
```

Zad. 5. (20 pkt. = 5*4)
Obok każdej instrukcji "cout", podaj
tekst, który zostanie wypisany. Wpisz
BŁĄD jeśli nie można tego jednoznacznie
określić lub gdy instrukcja powoduje błąd
wykonania programu.

```
struct A {  
    int y[1];  
    int *z;  
} s, *p, q;  
  
void s1( struct A s ) { *(s.y) = 5; }  
void s2( struct A s ) { *(s.z) = 6; }  
void s3( struct A *s ) { *(s->y) = 7; }  
  
void main() {  
    s.z = (int *) malloc( sizeof( int ) );  
    p = &s;  
    s1( s );  
    s2( s );  
    cout << s.y[0];  
    _____  
  
    cout << s.z[0];  
    _____  
  
    s3( p );  
    cout << s.y[0];  
    _____  
  
    cout << s.z[0];  
    q = s;  
    free( q.z );  
    cout << s.z[0];  
    return 0;  
}
```

Imię i nazwisko

Wypełnij drukowanymi literami

Numer indeksu

czas pisania: 75 minut, 23 styczeń 2015

Uwaga: w przypadku wszystkich programów należy założyć, że dołączone są biblioteki `iostream` i `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.

Zad. 1. (18 pkt. = 3*6) Podaj wartości wybranych elementów tablicy `liczby` tuż po zakończeniu realizacji funkcji `pytanie1` w podanym programie.

Odpowiedź:`liczby[0] = _____``liczby[1] = _____``liczby[2] = _____`

```
int liczby[] = { 7, 5, 11, 6, 4, 7 };
int pytanie1( int t[], int l, int r ) {
    if ( l < r ) {
        t[l] = t[r] - pytanie1(t+l/2, l+1, r/2) % 3;
    }
    return t[(l+r)/2];
}
int main() {
    pytanie1(liczby, 0, 5);
    return 0;
}
```

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

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

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

```
void pytanie2( char *napis ) {
    int a, b;
    for ( b=0; napis[b] != '\0'; _____ );

    for ( a=0; a < b/2; a++ ) {

        int c = napis[a], x = _____;

        napis[_____] = napis[x];
        napis[x] = c;
    }
}
int main() {
    char napis[] = { "WETIPG" };

    pytanie2( _____ );
    cout << napis;
    return 0;
}
```

Zad. 3. (25 pkt. = 5*5)

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.

```
void f( char *t ) { t[1] = 'E'; }
void g( char t ) { t = 'D'; }
```

```
int main() {
    int a[] = { 1, 2, 3, 5 };
    int *ptr = &(a[3]);
    char text[] = { "xyz" };
```

```
    cout << (int) (9/2/2.0);
```

```
    g( text[1] );
    cout << text;
```

```
    f( text + 2 );
    cout << sizeof( text );
```

```
    cout << (++ptr) - a;
```

```
    cout << text;
    return 0;
```

```
}
```

Zad. 4. (17 pkt. = 5+6+6)
Jaka wartość zostanie
zwrócona przez poniższe
wywołania funkcji:

pytanie4(5,0) _____

pytanie4(5,-1) _____

pytanie4(5,2) _____

```
int sgn( int x ) {  
    return ((x > 0) ? x : -x);  
}  
  
int pytanie4( int a, int b ) {  
    if ( a <= b || b == 0 )  
        return b-a;  
    else  
        return pytanie4( b, pytanie4( a, sgn(a-b+2)%sgn(b) ) );  
}
```

Zad. 5. (20 pkt. = 5*4)
Obok każdej instrukcji "cout", podaj
tekst, który zostanie wypisany. Wpisz
BŁĄD jeśli nie można tego jednoznacznie
określić lub gdy instrukcja powoduje błąd
wykonania programu.

```
struct A {  
    int y[1];  
    int *z;  
} s, *p, q;  
  
void s1( struct A s ) { *(s.z) = 5; }  
void s2( struct A s ) { *(s.y) = 6; }  
void s3( struct A *s ) { *(s->y) = 4; }  
  
int main() {  
    s.z = (int *) malloc( sizeof( int ) );  
    p = &s;  
    s1( s );  
    s2( s );  
    cout << s.y[0];  
    _____  
  
    cout << s.z[0];  
    _____  
  
    s3( p );  
    cout << s.y[0];  
    _____  
  
    cout << s.z[0];  
    q = s;  
    free( q.z );  
    cout << s.z[0];  
    return 0;  
}
```