Podstawy Programowania Komputerów

Wskaźniki

15 listopada 2018

1. Proszę zadeklarować wskaźnik na zmienną typu **int**.
2. W programie została zadeklarowana funkcja: **bool** `is_good`();. Proszę zadeklarować wskaźnik na tę funkcję.
3. W programie została zadeklarowana funkcja: **void** `revert`(**int** `tab`[ ], **const int** `SIZE`);. Proszę zadeklarować wskaźnik na tę funkcję.
4. Proszę zadeklarować tablicę `N`-elementową wskaźników na funkcję z punktu 3.
5. Proszę zadeklarować funkcję przyjmującą rozmiar tablicy i tablicę zadeklarowaną w punkcie 4.
6. Proszę przeczytać deklaracje:

```
const int N = 10;
int * a;
int b[N];
int c ();
int d (int a);
int e (int a[]);
int f (int);
int g (int []);
int h (int * a);
int i (int * a[]);
int * j (int);
int (* k) (int);
int * (* l) (int);
int (* m[N]) (int);
int (* n) (int (* f) (int));
int (* o) (int (* g[]) (int), const int);
int (* p[N]) (int (* g[]) (int), const int);
int (* q[N]) (int * (* g[]) (int), const int);
int (* r) (int (*) (int (*)(int)));
int * (* s) (int * (*) (int * (*)()));
int * (* t[N]) (int * (*) (int * (*)()));
int * (* u[N]) (int * (*[]) (int * (*)()));
int * (* v[N]) (int * (*[]) (int * (*[])()));
int * (* w[N]) (int * (*[]) (int * (*[])(int)));
int * (* x[N]) (int * (*[]) (int * (*[])(int [])));
int * (* y[N]) (int * (*[]) (int * (*[])(int * )));
int * (* z[N]) (int * (*[]) (int * (*[])(int * [])));
int (* A (char))(int);
int (* B (int))(char);
int (* C (char))(int, int *);
int (* D (char, char))(int, int *);
int * (* E (char, char))(int, int *);
```

```
int (* (* F) ())();
int (* (* G) (char))(int) = A;
int (* (* H) (int)) (char) = B;
int (* (* I) (char))(int);
int (* (* J) (int (* f) (int)))(int (* f) (int));
int (* (* K [N]) (int (* f) (int)))(int (* f) (int));
int (* (* L [N]) (int (*) (int)))(int (*) (int));
```

---

7. Proszę zaimplementować funkcje:

```
/** Proszę zaimplementować dowolną funkcję jednej zmiennej. */
double f (double x);

/** Funkcja całkująca numerycznie metodą prostokątów.
@param a dolna granica całkowania
@param b górna granica całkowania
@param dx krok całkowania
@param pf wskaźnik na funkcję do scałkowania
 */
double calkuj_prostokaty (double a, double b, double dx,
                          double (*pf) (double) );

/** Funkcja całkująca numerycznie metodą trapezów.
@param a dolna granica całkowania
@param b górna granica całkowania
@param dx krok całkowania
@param pf wskaźnik na funkcję do scałkowania
 */
double calkuj_trapezy (double a, double b, double dx,
                       double (*pf) (double) );

/** Funkcja całkująca numerycznie metodą parabol (Simpsona).
@param a dolna granica całkowania
@param b górna granica całkowania
@param dx krok całkowania
@param pf wskaźnik na funkcję do scałkowania
 */
double calkuj_parabole (double a, double b, double dx,
                        double (*pf) (double) );

/** Funkcja wyznaczająca najdokładniejszą metodę całkowania. Funkcja
    otrzymuje tablicę wskaźników do funkcji całkujących, a następnie
    sprawdza, która funkcja uzyskuje najdokładniejszy wynik i zwraca
    wskaźnik na tę najdokładniejszą funkcję.
@param a dolna granica całkowania
@param b górna granica całkowania
@param dx krok całkowania
@param cel dokładna wartość całki
@param tabpf tablica wskaźników na funkcje całkujące
@param SIZE rozmiar tablicy wskaźników na funkcje całkujące
@return wskaźnik na najdokładniejszą funkcję całkującą
 */
double (* najdokladniejsza (double a, double b, double dx, const double
    cel, double (*tabpf[]) (double, double, double, double (*) (double))
    , const int SIZE)) (double, double, double, double (*) (double));
```

```
// Zapis powyższej deklaracji może być trochę hermetyczny. Ale można zapisać to samo w nieco
    przyjaźniejszy sposób korzystając z typedef:


/** wskaźnik na funkcję jednej zmiennej */
typedef double (*pFunkcjaJednejZmiennej) (double);

/** wskaźnik na funkcję całkującą */
typedef double (*pFunkcjaCalkujaca) (double a, double b, double dx,
    pFunkcjaJednejZmiennej pf);

/** Funkcja wyznaczająca najdokładniejszą metodę całkowania. Funkcja
    otrzymuje tablicę wskaźników do funkcji całkujących, a następnie
    sprawdza, która funkcja uzyskuje najdokładniejszy wynik i zwraca
    wskaźnik na tę najdokładniejszą funkcję.
@param a dolna granica całkowania
@param b górna granica całkowania
@param dx krok całkowania
@param cel dokładna wartość całki
@param tabpf tablica wskaźników na funkcje całkujące
@param SIZE rozmiar tablicy wskaźników na funkcje całkujące
@return wskaźnik na najdokładniejszą funkcję całkującą  */
pFunkcjaCalkujaca najdokladniejsza (double a, double b, double dx, const
     double cel, pFunkcjaCalkujaca tabpf[], const int SIZE)
```