Řešení sady 3

Úvod do programování 1

Tomáš Kühr

Obrácení pole

```
double testovaci[VELIKOST];

// naplneni pole
for (int i = 0; i < VELIKOST; i++){
    testovaci[i] = 2 * i - 5;
}

// obraceni pole
for (int i = 0; i < VELIKOST/2; i++){
    double temp; // pomocna promenna
    temp = testovaci[i];
    testovaci[i] = testovaci[VELIKOST - i - 1];
    testovaci[VELIKOST - i - 1] = temp;
}

// vypis pole
for (int i = 0; i < VELIKOST; i++){
    printf("%g, ", testovaci[i]);
}
```

Eratosthenovo síto

```
int nevyskrtnuto[VELIKOST];

// naplneni pole
nevyskrtnuto[0] = nevyskrtnuto[1] = 0;
for (int i = 2; i < VELIKOST; i++) nevyskrtnuto[i] = 1;

for (int i = 0; i < VELIKOST; i++){ // pruchod polem
    if (nevyskrtnuto[i]){ // pro nevyskrtnuto cislo
        for (int j = 2 * i; j < VELIKOST; j += i){
            nevyskrtnuto[j] = 0; // skrtame jeho násobky
        }
    }
}

// vypis neskrtnutych
for (int i = 0; i < VELIKOST; i++){
    if (nevyskrtnuto[i]) printf("%d, ", i);
}
```

Průměr pole

```
double testovaci[VELIKOST];

// naplneni pole
for (int i = 0; i < VELIKOST; i++){
    testovaci[i] = 2 * i - 5;
}

// prumer pole
double prumer = 0;
for (int i = 0; i < VELIKOST; i++){
    prumer += testovaci[i];
}

prumer = prumer / VELIKOST;

printf("Prumer je %g.\n", prumer);
```

Suma pole

```
double suma(double cisla[], int pocet){
    double vysledek = 0;
    for (int i = 0; i < pocet; i++){
        vysledek += cisla[i];
    }
    return vysledek;
}

int main(){
    double testovaci[VELIKOST];
    ...
    // volani sumy pole
    double soucet;
    soucet = suma(testovaci, VELIKOST);
    printf("Suma je %g.\n", soucet);

    return 0;
}
```

Převody do desítkové soustavy

```
int do_desitkove(char cislo[], int zaklad){
    int vysledek = 0;
    int i = 0;
    char znak;
    while (znak = cislo[i]){
        int hodnota = -1;
        if ('0' <= znak && znak <= '9') hodnota = znak - '0';
        if ('A' <= znak && znak <= 'Z') hodnota = znak - 'A' + 10;
        if (hodnota == -1 || hodnota >= zaklad){
            printf("Chyba - neplatna cislice.\n");
            return -1;
        }
        vysledek = (vysledek * zaklad) + hodnota;
        i++;
    }
    return vysledek;
}
```

Hledání čísla v poli

```
const int velikost = 10;
double testovaci[velikost];
double hledane;
int indexHledaneho = -1;
...

for (int i = 0; i < velikost; i++){
   if (testovaci[i] == hledane){
      indexHledaneho = i;
      break;
   }
}

if (indexHledaneho==-1)
   printf("Cislo %g nebylo nalezeno. \n", hledane);
else
   printf("Cislo %g je na indexu %d. \n", hledane, indexHledaneho);
```

Počet čísel v intervalu

```
const int velikost = 10;
double testovaci[velikost];
double pocatek;
double konec;
int nalezeno = 0;

...

for (int i = 0; i < velikost; i++){
   if (pocatek <= testovaci[i] && testovaci[i] <= konec){
      nalezeno++;
   }
}

printf("V intervalu bylo nalezeno %d cisel. \n", nalezeno);
```

Objem a povrch - funkce

```
#include <math.h>

double objemValce(double r, double v){
    return M_PI * r * r * v;
}

double povrchValce(double r, double v){
    return 2 * M_PI * r * (r + v);
}

double objemHranolu(double podstava, double vyska){
    return podstava * podstava * vyska;
}

double povrchHranolu(double podstava, double vyska){
    return 2 * podstava * podstava + 4 * podstava * vyska;
}
```

Objem a povrch – použití funkcí

```
int typTelesa;
double prvni, druhy;

...

if (typTelesa == 1)
   printf("Valec - objem %g, povrch %g\n",
          objemValce(prvni, druhy), povrchValce(prvni, druhy));

if (typTelesa == 2)
    printf("Hranol - objem %g, povrch %g\n",
           objemHranolu(prvni, druhy), povrchHranolu(prvni, druhy));
```