BPC-UDP Cvičení 11

# Příklad 1

Upravte program z příkladu 5, cvičení 10 tak, aby vstupní čísla byla čtena z textového souboru a nalezená čísla s minimální a maximální hodnotou byla zapsána do textového souboru.

1. Založte projekt a přidejte do něj soubory main.c, check.h a check.c. Soubor main.c převezměte z příkladu 5, cvičení 10.
2. Vytvořte soubor *vstup.txt*. Soubor může mít např. následující obsah:  
   **10**  
   **7.0 -3.5 -8.2 4.3 7.1 10.0 12.0 -5.8 7.1 -3.5**  
   Číslo na prvním řádku v souboru (zde **10**) je typu **size\_t** a udává počet čísel typu **double** zapsaných na následujícím řádku, z nichž má program určit nejmenší a největší číslo.
3. Ve funkci main():
   1. Definujte proměnnou fr jako pointer na stream **FILE**. Otevřete vstupní soubor pojmenovaný *vstup.txt* pro čtení.  
      Pokud se otevření souboru nepodaří, vypište na standardní **chybový** výstup „*Chyba pri otevirani vstupniho souboru!*“ a program ukončete s návratovým kódem **2**.
   2. Přečtěte ze vstupního streamu fr číslo typu **size\_t** reprezentující počet čísel, která mají být zpracována, a uložte ho do proměnné data\_num.  
      Pokud při čtení dojde k chybě, uzavřete stream fr, vypište na standardní **chybový** výstup „*Chyba pri cteni poctu dat!*“ a program ukončete s návratovým kódem **3**.
   3. Dynamicky alokujte pole pro uložení zadaného počtu čísel typu **double**. Na začátek pole bude ukazovat pointer ptr\_data.  
      Pokud by alokace selhala, uzavřete stream fr, vypište na standardní chybový výstup chybovou hlášku „*Chyba alokace!*“ a program ukončete s návratovým kódem **4**.
   4. Čtěte ze vstupního streamu fr čísla typu **double** a ukládejte je do dynamicky alokovaného pole, na jehož první element ukazuje pointer ptr\_data. Počet čísel, která je třeba přečíst, je dán hodnotou proměnné data\_num.  
      Pokud při čtení dojde k chybě, uvolněte dynamicky alokovanou paměť, uzavřete stream fr, vypište na standardní **chybový** výstup „*Chyba pri cteni dat!*“ a program ukončete s návratovým kódem **5**.
   5. Uzavřete stream fr.
   6. Vypište na standardní výstup čísla uložená v poli, na jehož první element ukazuje pointer ptr\_data.
   7. Zavolejte funkci minmax().
   8. Pokud funkce minmax() vrátila hodnotu **-1**, uvolněte dynamicky alokovanou paměť, uzavřete stream fr a program ukončete s návratovým kódem **6**.
   9. Uvolněte dynamicky alokovanou paměť.
   10. Vytiskněte na standardní výstup hodnoty proměnných min a max.
   11. Definujte proměnnou fw jako ukazatel na stream **FILE**. Otevřete výstupní soubor pojmenovaný *vystup.txt* pro zápis.  
       Pokud se otevření souboru nepodaří, vypište na standardní **chybový** výstup „*Chyba pri otevirani vystupniho souboru!*“ a program ukončete s návratovým kódem **7**.
   12. Zapište do výstupního streamu fw minimální a maximální hodnotu (obsah proměnných min a max) s doprovodným textem „*min=*“ a „*max=*“. Každou hodnotu zapište na samostatný řádek.  
       Pokud při zápisu do výstupního souboru dojde k chybě, uzavřete výstupní soubor a vypište na standardní **chybový** výstup „*Chyba pri zapisu do vystupniho souboru!*“ a program ukončete s návratovým kódem **8**.
   13. Uzavřete stream fw a program ukončete s návratovým kódem **0**.

# Příklad 2

Upravte program z příkladu 1 tak, aby jméno vstupního souboru bylo zadáno jako první parametr příkazového řádku a jméno výstupního souboru jako druhý parametr příkazového řádku.  
Pokud uživatel nezadá alespoň dva parametry příkazového řádku, vypište na standardní **chybový** výstup chybovou hlášku „*Chyba – nebylo zadano jmeno vstupniho a vystupniho souboru!*“ a program ukončete s návratovým kódem **1**.

# Příklad 3

Upravte program z příkladu 2 tak, aby počet zpracovávaných čísel (v příkladu 2 proměnná data\_num) a ukazatel na pole čísel (v příkladu 2 proměnná ptr\_data) byly zapouzdřeny ve struktuře TData.

1. Definujte strukturu TData, která bude mít složky:  
   - iPtr\_data typu pointer na double **double**,  
   - iData\_num typu **size\_t**.
2. Upravte deklaraci (úplný funkční prototyp) funkce minmax(), která nalezne v poli čísel nejmenší a největší číslo. Funkce bude mít 3 formální parametry:  
   - aData typu struktura TData,  
   - aPtrMin a aPtrMax ukazatele na proměnné typu **double**.  
   Návratová hodnota funkce bude typu **int**.
3. Upravte definici funkce minmax().
4. Ve funkci main():
   1. Zrušte definici proměnných data\_num a ptr\_data.
   2. Definujte proměnnou data typu struktura TData.
   3. Upravte zbytek kódu tak, aby využíval proměnnou data.

# Příklad 4

Napište a odlaďte program v jazyce C, který přečte ze vstupního souboru data popisující souřadnice bodů v rovině a uloží je do dynamicky alokovaného pole struktur. Následně informace o souřadnicích bodů vytiskne na standardní výstup.

1. Založte projekt a přidejte do něj soubory main.c, check.h a check.c. Soubor main.c můžete převzít z příkladu 2.
2. Vytvořte soubor *vstup.txt*. Soubor může mít např. následující obsah:  
   **5**  
   **7.0 -3.5  
   -8.2 4.3  
   7.1 10.0  
   12.0 -5.8  
   7.1 -3.5**  
   Číslo na prvním řádku v souboru (zde **5**) je typu **size\_t** a udává počet bodů. Na následujících řádcích jsou dvojice čísel typu **double**. První číslo reprezentuje X-ovou souřadnici bodu, druhé číslo Y-ovou souřadnici bodu.
3. Do souboru main.c, vložte následující definici struktury TPoint:  
   **struct** TPoint { **double** iX; **double** iY; };
4. Definujte funkci main. Ve funkci main:
   1. Definujte proměnnou points\_ptr typu pointer na strukturu TPoint.
   2. Definujte proměnnou fr jako ukazatel na stream **FILE**. Otevřete vstupní soubor pojmenovaný *vstup.txt* pro čtení.  
      Pokud se otevření souboru nepodaří, vypište na standardní **chybový** výstup „*Chyba pri otevirani vstupniho souboru!*“ a program ukončete s návratovým kódem **2**.
   3. Přečtěte ze vstupního streamu fr číslo typu **size\_t** reprezentující počet bodů (tedy struktur TPoint), které mají být zpracovány, a uložte ho do proměnné points\_num.  
      Pokud při čtení dojde k chybě, uzavřete stream fr, vypište na standardní **chybový** výstup „*Chyba pri cteni poctu dat!*“ a program ukončete s návratovým kódem **3**.
   4. Dynamicky alokujte pole pro uložení zadaného počtu struktur TPoint. Na první element pole bude ukazovat pointer points\_ptr.  
      Pokud alokace selhala, uzavřete stream fr, vypište na standardní chybový výstup chybovou hlášku „*Chyba alokace!*“ a program ukončete s návratovým kódem **4**.
   5. Čtěte ze vstupního streamu fr souřadnice jednotlivých bodů a ukládejte je do elementů pole, na jehož začátek ukazuje points\_ptr. (Elementem pole je struktura TPoint, souřadnice je třeba ukládat do složek iX, iY.) Počet elementů pole, které je třeba naplnit, je dán hodnotou proměnné points\_num.  
      Pokud při čtení dojde k chybě, uvolněte dynamicky alokovanou paměť, uzavřete stream fr, vypište na standardní **chybový** výstup „*Chyba pri cteni dat!*“ a program ukončete s návratovým kódem **5**.
   6. Uzavřete stream fr.
   7. Vypište na standardní výstup souřadnice bodů uložených v poli, na které ukazuje points\_ptr a jejich vzdálenost od počátku souřadnic. Informace o každém bodu tiskněte na samostatný řádek.
   8. Uvolněte dynamicky alokovanou paměť a program ukončete s návratovým kódem **0**.

# Příklady pro samostatné procvičení probrané látky

# Příklad 5

Upravte program z přikladu 4 tak, aby tisk souřadnic bodů byl realizován ve funkci:   
**void** points\_print(**size\_t** aPointsNum, **const** struct TPoint aPointsPtr[]);

Napište definici funkce points\_print().

V bodě g) zavolejte funkci points\_print().

# Příklad 6

Do programu z příkladu 4 přidejte deklaraci a definici funkce:  
**struct** TPoints points\_max\_distance(**size\_t** aPointsNum, **const** **struct** TPoint aPointsPtr[]);

Funkce points\_max\_distance() nalezne bod, který má největší vzdálenost od počátku souřadnic. Pro výpočet vzdálenosti bodu využijte funkci distance() z příkladu 3, cvičení 9.