INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

DUM 3 téma: Nejmenší ze tří

ze sady: 1 **tematický okruh sady:** Algoritmy a datové struktury
ze šablony: 10 – Algoritmizace a programování **určeno pro:** 1. a 2. ročník
vzdělávací obor: 18-20-M/01 Informační technologie
26-41-M/01 Elektrotechnika - Elektronické počítačové systémy
vzdělávací oblast: odborné vzdělávání
metodický list/anotace: VY_32_INOVACE_10103ml.pdf
pomocné soubory: nejmensi1.cpp, nejmensi2.cpp, nejmensi3.cpp, nejmensi4.cpp

Vývojové diagramy slouží k (doplňte) .....

I. Nejmenší prvek ze tří – jednoduchý algoritmus

Máme tři čísla uložená v proměnných ***a***, ***b***, ***c***.
Pokud je nejmenší ***a***, musí platit následující podmínky: _____
Pokud je nejmenší ***b***, musí platit následující podmínky: _____
Pokud je nejmenší ***c***, musí platit následující podmínky: _____

a. Doplňte program (vnitřek funkce main):

```
int a,b,c;
scanf("%d %d %d",&a,&b,&c);
if(.................)
  printf("%d",......);
if(.................)
  printf("%d",......);
if(.................)
  printf("%d",......);
```

b. Vytvořte příslušný vývojový diagram:

c. Počet porovnání

Počet porovnání v programu: _____
Při každém spuštění programu se provede _____ porovnání.
Vyznačte tato porovnání ve vývojovém diagramu.

II. Nejmenší prvek ze tří – efektivní algoritmus

a. Správnost algoritmu

Algoritmus je správný, pokud
1. se algoritmus zastaví a
2. počítá to, co má.

b. Náročnost algoritmu

Zjišťujeme, kolik program spotřebuje:

1. ļasu procesoru (poļítáme poļet operací, zejména tŋch náročnŋjších) a
2. pamŋti (chápejme tím operaļní pamŋť).

c. Lepší algoritmus (Nejmenší prvek ze tří)

Chceme zredukovat poļet porovnání, které se provede v prŧmŋru při každém spuštŋní programu. Upravte předchozí vývojový diagram tak, aby se druhá a třetí podmínka nacházela v ***else*** vŋtvích předchozích podmínek:

Poļet porovnání v programu:
_____

Při každém spuštŋní programu se provede v prŧmŋru _____ porovnání.

d. Efektivní algoritmus

V předchozím případŋ se nŋkteré podmínky provádŋjí vícekrát. Pokud je nejmenší ļíslo ***c***, tak ho ___krát porovnáváme s ļíslem ***a*** a ___krát porovnáváme s ļíslem ***b***.
V následujícím vývojovém diagramu odstraníme tato násobná porovnání:

III. Cviļení

1. Na základŋ vývojového diagramu vytvořte příslušný program.
2. Nakreslete vývojový diagram algoritmu, který vypoļítá mocninu 2^n. Ļíslo ***n*** uživatel zadá z klávesnice.
3. Co poļítá program daný následujícím vývojovým diagramem. Kolik se maximálnŋ provede porovnání? Kolik je to v prŧmŋru (přibližnŋ)? Jak lze snadno program upravit a snížit tak poļet opakování (aby program poļítal stále to samé)?