

Úkolem je realizovat program, který bude počítat zisk a ztrátu při obchodování s akcemi.

Předpokládáme, že zpracováváme cenu akcie v čase. Cenu akcie se postupně dozvídáme pro jednotlivé časové okamžiky. Pro známé hodnoty ceny v minulosti chceme vypočítat maximální možný zisk a maximální možnou ztrátu. Při výpočtu dostaneme zadaný časový interval, ve kterém obchod proběhne. Akcii můžeme jednou nakoupit (někdy v zadaném časovém intervalu, za cenu platnou v okamžiku nákupu) a jednou prodat (opět v zadaném časovém intervalu, za cenu platnou v dobu prodeje). Při obchodování nelze shortovat, tj. musíme nejprve akcii nakoupit a teprve pak prodat.

Vstupem programu je zadání ceny akcie prokládané dotazy na možný zisk a ztrátu. Cena akcie je zadána ve tvaru + cena, např. + 200. Dotaz na zisk/ztrátu má podobu ? from to, např. ? 5 10. Ceny akcie a dotazy jsou zadávané postupně, zadávání končí dosažením konce vstupu (EOF). Následuje ukázkový běh #1 s dodaným vysvětlením (za znakem #):

```
+ 20          # cena v čase 0
+ 30          # cena v čase 1
+ 40          # cena v čase 2
+ 10          # cena v čase 3
? 0 2         # hledání max. zisku/ztráty pro období <0;2>
? 0 3         # hledání max. zisku/ztráty pro období <0;3>
+ 50          # cena v čase 4
? 0 3         # hledání max. zisku/ztráty pro období <0;3>
? 1 4         # hledání max. zisku/ztráty pro období <1;4>
```

Výstupem programu je řešení jednotlivých zadaných problémů. Pro každý zadaný problém je zobrazena informace o maximálním možném zisku (spolu s časovým okamžikem nákupu a prodeje akcie) a informace o maximální možné ztrátě (opět s časovými okamžiky nákupu a prodeje). Pokud v zadaném intervalu nelze realizovat zisk/ztrátu, je výsledkem hodnota N/A, viz ukázka.

Pokud vstup není platný, program tuto situaci detekuje, vypíše chybové hlášení a ukončí se. Formát chybového hlášení je opět uveden v ukázkách níže. Za chybu je považováno:

- nerozpoznaný vstup (nezačíná znakem + ani ?),
- neplatné zadání ceny akcie (nečíselné, záporná cena),
- neplatné zadání dotazu (neobsahuje dvě celá čísla, počátek intervalu je záporný, konec intervalu je větší nebo roven počtu dosud známých cen akcií, počátek intervalu je větší než konec intervalu).

Pokud program detekuje chybu, přestane se dotazovat na další vstupní hodnoty, vypíše chybové hlášení a ukončí se. Chybu je tedy potřeba detekovat okamžitě po načtení hodnoty (neodkládejte kontrolu vstupních údajů až za načtení celého vstupu). Chybové hlášení vypisujte na standardní výstup (nevypisujte jej na standardní chybový výstup).

Dodržte přesně formát všech výpisů. Výpis Vašeho programu musí přesně odpovídat ukázkám. Testování provádí stroj, který kontroluje výpis na přesnou shodu. Pokud se výpis Vašeho programu liší od referenčního výstupu, je Vaše odpověď považována za nesprávnou. Záleží i na mezerách, i na odřádkování. Nezapomeňte na odřádkování za posledním řádkem výstupu (a

za případným chybovým hlášením). Využijte přiložený archiv s testovacími vstupy a usnadněte si testování Vašeho programu.

Váš program bude spouštěn v omezeném testovacím prostředí. Je omezen dobou běhu (limit je vidět v logu referenčního řešení) a dále se kontroluje i velikost použité paměti (k dispozici je dostatek paměti pro uložení cen akcií a dalších vypočtených hodnot; paměťové požadavky však nesmí nesmyslně překračovat rozumné meze). Zadaný problém lze řešit algoritmy s různou efektivitou:

- Velmi neefektivní naivní algoritmus zvládne povinné testy, ale neprojde testem nepovinným (bude hodnocen méně než 100% bodů). Řešení jednoho problému může trvat i déle než lineární čas (vzhledem k velikosti zadného intervalu).
- Rozumně implementovaný a vylepšený naivní algoritmus projde povinnými testy, ale nezvládne časový limit testu bonusového. Očekává se, že jeden zadaný problém bude vyřešen v lineárním čase vzhledem k velikosti zadaného intervalu. Takové řešení bude hodnoceno 100% bodů.
- Pro zvládnutí bonusového testu je potřeba efektivnější algoritmus, který dokáže rychle zpracovat dlouhé časové intervaly a velký počet dotazů. Očekává se, že jeden zadaný problém bude vyřešen významně rychleji než v lineárním čase vzhledem k velikosti zadaného intervalu.

#### **Ukázka práce programu:**

---

**Ceny, hledani:**

+ 20

+ 30

+ 40

+ 10

? 0 2

**Nejvyšší zisk: 20 (0 - 2)**

**Nejvyšší ztrata: N/A**

? 0 3

**Nejvyšší zisk: 20 (0 - 2)**

**Nejvyšší ztrata: 30 (2 - 3)**

+ 50

? 0 3

**Nejvyšší zisk: 20 (0 - 2)**

**Nejvyšší ztrata: 30 (2 - 3)**

? 1 4

**Nejvyšší zisk: 40 (3 - 4)**

**Nejvyšší ztrata: 30 (2 - 3)**

---

**Ceny, hledani:**

+ 100

+ 110

+ 105

+ 120

+ 70

+ 105

+ 50

+ 85

+ 40

? 0 8

Nejvyšší zisk: 35 (4 - 5)

Nejvyšší ztrata: 80 (3 - 8)

? 0 6

Nejvyšší zisk: 35 (4 - 5)

Nejvyšší ztrata: 70 (3 - 6)

? 3 5

Nejvyšší zisk: 35 (4 - 5)

Nejvyšší ztrata: 50 (3 - 4)

? 2 8

Nejvyšší zisk: 35 (4 - 5)

Nejvyšší ztrata: 80 (3 - 8)

? 7 7

Nejvyšší zisk: N/A

Nejvyšší ztrata: N/A

---

Ceny, hledání:

+ 10

+ 20

+ 30

+ 10

+ 20

+ 30

+ 5

+ 25

? 0 7

Nejvyšší zisk: 20 (0 - 2)

Nejvyšší ztrata: 25 (2 - 6)

? 2 7

Nejvyšší zisk: 20 (3 - 5)

Nejvyšší ztrata: 25 (2 - 6)

? 4 7

Nejvyšší zisk: 20 (6 - 7)

Nejvyšší ztrata: 25 (5 - 6)

+ 50

? 0 8

Nejvyšší zisk: 45 (6 - 8)

Nejvyšší ztrata: 25 (2 - 6)

---

Ceny, hledání:

+ 10

\* 20

Nesprávný vstup.

---

Ceny, hledání:

+ 10

+ 20

+ 30

? 0 2

Nejvyšší zisk: 20 (0 - 2)

Nejvyšší ztrata: N/A

**Poznámky:**

- Ukázkové běhy zachycují očekávané výpisy Vašeho programu (tučné písmo) a vstupy zadané uživatelem (základní písmo). Zvýraznění tučným písmem je použito pouze zde na stránce zadání, aby byl výpis lépe čitelný. Váš program má za úkol pouze zobrazit text bez zvýrazňování (bez HTML markupu).
- Znak odřádkování (`\n`) je i za poslední řádkou výstupu (i za případným chybovým hlášením).
- Program si část vstupních dat musí uložit do paměti, ideálně do pole. Velikost vstupu není explicitně omezena, pro testy s malým objemem vstupních dat je nastaven restriktivní paměťový limit. Data budete muset alokovat dynamicky.
- Povinné testy rychlosti zadávají cenu akcií pro dlouhý časový interval (stovky tisíc hodnot). Zisk/ztráta se vyhledává v krátkých intervalech, tedy algoritmus výpočtu nemusí být úplně efektivní. Nepovinný test vyhledává v delších intervalech. Konečně, bonusový test vyhledává ve velmi dlouhých intervalech a vyhledávání je zadáváno velmi mnoho. Navíc se prokládají zadávání cen akcií a vyhledávání.
- Pro některé dotazy může existovat více správných řešení. Například v ukázkovém běhu #3 pro dotaz ? 0 7 lze dosáhnout maximálního zisku 20 pro obchody: 0 – 2, 3 – 5, 0 – 5 a 6 – 7 (čísla udávají okamžik nákupu a prodeje). Pokud existuje více možností, program zobrazí výsledek s co nejmenším dnem nákupu (zde 0 – 2 a 0 – 5). Pokud stále existuje více možností, program zobrazí variantu s co nejmenším dnem prodeje (zde 0 – 2).
- Správné řešení této úlohy, které zvládne povinné a nepovinné testy na 100%, lze použít pro code review (tj. pro code review nemusíte zvládnout bonusový test).