

# Korespondenční Seminář z Programování

## ZAČÁTEČNICKÁ KATEGORIE

26. ročník

KSP-Z

Listopad 2013

Právě se vám do rukou dostává historicky první zadání začátečnické kategorie KSP. Ta by oproti klasickému KSP měla být mnohem snazší a přístupná třeba i lidem, kteří se doposud s programováním a informatikou potkali jen velmi letmo.

KSP-Z se snaží být odrazovým můstkem pro každého, kdo se chce naučit používat počítač nejen ke čtení emailů a je zvědavý na taje informatiky.

Budete v něm řešit úlohy jak ryze praktické, kde vám budete posílat jen výsledky na zadané vstupy, tak úlohy teoretické, kde bude naopak záležet hlavně na návrhu a popisu algoritmu. Výsledek praktických úloh zjistíte ihned, teoretické úlohy vám opravíme po skončení série. Ke všem úlohám zveřejníme také vzorová řešení, ze kterých máte možnost se poučit. Pak na jaře ty nejuspěšnější z vás pozveme na týdenní soustředění v přírodě, kde své schopnosti ještě zdokonalíte a zároveň si užijete spoustu zábavy.

Termín odevzdání první série je stanoven na **pondělí 6. ledna 2014 v 8:00 SEČ**.

Řešení přijímáme elektronicky na <https://ksp.mff.cuni.cz/z/>. Tam také můžete nalézt další informace o tom, jak KSP a KSP-Z fungují. Rovněž tam najdete fórum, kde se můžete na cokoli zeptat. Anebo nám můžete napsat na email [ksp@mff.cuni.cz](mailto:ksp@mff.cuni.cz).



### Zadání první série začátečnické kategorie 26. ročníku KSP

První čtyři úlohy (označené piktogramem počítače) jsou praktické. To znamená, že ke každé z nich máme na webu ke stáhnutí několik sad vstupů a body získáš za odevzdání správných výstupů k odpovídajícím sadám. Jak výstupy vygeneruješ (třeba v jakém jazyce napíšeš program, který je zpracuje), je už čistě na Tobě.

#### 26-Z1-1 Kevin a magnety 8 bodů

Kevin má v krabičce  $N$  magnetů – obyčejných magnetických tyček, které mají na jednom konci pól  $+$  a na druhém  $-$ . Jednoho dne se rozhodl, že si v nich udělá pořádek.

Jeden po druhém magnety vyndával z krabičky a pokládal je do řady na stůl. Někdy se spojily k sobě (pokud se setkaly opačnými póly), jindy se mezi nimi zase vytvořila mezera.

Kevinu by zajímalo, do kolika částí se mu magnety spojily.

**Tvar vstupu:** Program na vstupu dostane na prvním řádku číslo  $N$  ( $1 \leq N \leq 1000000$ ). Každý z dalších  $N$  řádků bude obsahovat buď řetězec  $+-$ , nebo  $-+$  udávající, kterým směrem Kevin magnet položil.

**Tvar výstupu:** Na výstup vypíše jediný řádek obsahující jedno celé číslo: počet oddělených částí, do kterých se magnety spojí.

**Ukázkový vstup:** **Ukázkový výstup:**

```
4          3
+-
+-
-+
+-
```

#### 26-Z1-2 Piškvorky 10 bodů

Kevin sedí na hodině dějepisu. Bitvy, letopočty, děsná nuda. Raději si hraje se svým novým smartphonem a píše si s kamarádkou Sárrou, která se ve vedlejší třídě nudí na hodině matiky. Aby si zkrátili dlouhou chvíli, rozhodli se hrát piškvorky.

Jejich aplikace na piškvorky ale není dokonalá a neumí poznat, jestli už někdo vyhrál. To si Kevin se Sárrou uvědomili

až po nějaké době, a tak by je nyní zajímalo, kolikrát kdo z nich vyhrál. Kevin má křížky a Sára kolečka.

**Tvar vstupu:** Na vstupu dostanete čísla  $S$  a  $R$  udávající šířku a výšku hrací plochy ( $1 \leq R, S \leq 1000$ ). Na dalších  $R$  řádcích bude vždy  $S$  znaků  $X$ ,  $O$ , nebo  $.$ , kde  $X$  znamená křížek,  $O$  kolečko a  $.$  znamená prázdné políčko hrací plochy.

**Tvar výstupu:** Na výstup vypíše jediný řádek se dvěma celými čísly oddělenými mezerou: počtem pětic křížků a počtem pětic koleček. Pětice mohou ležet v libovolném řádku, sloupci, nebo úhlopříčce.

**Ukázkový vstup:** **Ukázkový výstup:**

```
6 6          4 3
XXXXXX
.X. .OX
..XO.X
..OX.X
.O..XX
00000X
```

#### 26-Z1-3 Zamilovaný dopis 10 bodů

Kevin působí jako bubeník ve studentské rockové kapelě *Velká tlama*. Kapela během svého prvního veřejného koncertu sklídila ohromný úspěch. Další den Kevin objevil ve schránce  $K$  dopisů. Mezi všemi nudnými úředními dopisy jeden vyčníval. Byl navoněný a na obálce měl nakreslené růžové srdíčko. Ten si okamžitě přečetl. Byl od fanynky!



Takové dopisy Kevin nedostává každý den. Obsah dopisu si tak hned slovo od slova zapamatoval. Bohužel pak při cestě

do školy přešlo a dopisy Kevinovi zmokly. Některé se staly téměř nečitelnými. I tak se ale snažil a přečetl z nich, co jen šlo. Posloupností těchto znaků si zapsal. Nyní by chtěl vědět, který z těchto dopisů by mohl být oním dopisem se srdíčkem.

**Tvar vstupu:** Na prvním řádku vstupního souboru bude zadán text původního dopisu, jehož délka bude nejvýše 300 000 znaků. Na druhém řádku bude zadáno číslo  $K$  udávající celkový počet doručených dopisů ( $1 \leq K \leq 50$ ). Na každém ze zbylých  $K$  řádků bude posloupnost maximálně 300 000 znaků, které se Kevinovi z daného dopisu podařilo přečíst. Všechny znaky dopisů budou složeny z malých a velkých písmen anglické abecedy a znaků  $.$ ,  $,$ ,  $_$ , kde  $_$  odděluje jednotlivá slova.

**Tvar výstupu:** Vypíše  $K$  řádků výstupu. Na  $i$ -tý řádek umístíte řetězec ANO, pokud  $i$ -tý dopis může být oním dopisem se srdíčkem. To znamená, že  $i$ -tý dopis lze z původního dopisu vytvořit vynecháním některých znaků. V opačném případě vypíše řetězec NE.

**Ukázkový vstup:** **Ukázkový výstup:**

```
Horoucne_Te_miluji,_Kevine.  NE
5                             ANO
Horoucne_Te_milujeme,_Kevine.  NE
nemiluji                       ANO
MILUJI.                         ANO
Hor___eve.
Horce_Te_miluji,Kene.
```

#### 26-Z1-4 Hroch v jezeře 12 bodů

Kevin po odpoledních hraje počítačovou hru Hroch v jezeře. Ve hře plave s hrochem ve velkém jezeře, ze kterého se tu a tam vynořuje ostrůvek. Některé ostrůvky jsou pusté, na jiných se nachází hroší laskominy – jídlo všeho druhu. Cílem hry je doplatit s hrochem na ostrůvek s co největším množstvím jídla a mezitím nevyležat z vody.

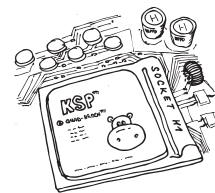
Ostrov je souvislá oblast políček pevniny, kde políčka považujeme za spojená, pokud se dotýkají hranou (nikoliv pouze rohem). Hroch může plavat nahoru, dolů, doleva a doprava.

**Tvar vstupu:** Na prvním řádku vstupu dostanete čísla  $S$  a  $R$  udávající šířku a výšku mapy hry. Na každém z dalších  $R$  řádků bude  $S$  znaků, přičemž:

- $.$  znamená, že na políčku je voda,
- $\#$  znamená, že na políčku je pevnina bez jídla,
- $J$  je pevnina s jídlem,
- $H$  je políčko, kde hroch začíná.

Zaručujeme, že na vstupu bude právě jedno políčko se znakem  $H$ .

**Tvar výstupu:** Na výstup napíše jedině celé číslo udávající největší množství jídla na ostrově, ke kterému se hroch může ze své startovní pozice doplatit povolenými způsoby bez toho, aby mezitím vystoupil z vody.



**Ukázkový vstup:**

```
10 8
.J##..#.J#
.###.J#.J
#JJJ#..#.J
..###.J.J
.....#.J
.J.H.###.J
.J...#JJ.#
.J#..#JJ.J
```

**Ukázkový výstup:**

6

Zbylé dvě úlohy této série jsou teoretické. K těmto úlohám je třeba vymyslet algoritmus a slovně ho popsat – tedy primární část řešení není zdrojový kód v nějakém programovacím jazyce, ale slovní popis. Sepsanou úlohu odevzdej opět na našem webu.

#### 26-Z1-5 Úkol z geometrie 12 bodů

Kevin a jeho spolužáci dostali na hodině geometrie následující úkol. Mají zadaných  $N$  bodů na ose  $x$ . Je třeba každé dva po sobě jdoucí body spojit horní půlkružnicí. Kevin se podíval na posloupnost ze zadání a začal přemýšlet, jestli se některé dvě půlkružnice protnou. Navrhnete co neefektivnější algoritmus, který pro zadanou posloupnost  $N$  bodů odpoví ANO, pokud se některé půlkružnice protínají, a NE, pokud nikoliv.

Na prvním obrázku je příklad pro posloupnost 0, 8, 20, 14, 18 a na druhém pro posloupnost 0, 12, 4, 10, 20. V prvním se kružnice neprotínají, v tom druhém protínají.



#### 26-Z1-6 Nezbední skřítky 14 bodů

Kevin byl na vánočních trzích a co nesehnal – opravdové, živé vánoční skřítky! Tomu nemohl odolat a hned si jich  $N$  koupil. Dokonce byl každý jinak velký.

Aby mu neutekli, vyrobil si pro ně doma řadu  $N$  klíček a každého do jedné zavleč. Zavření skřítky ale dělali příšerný kravál a neustále si stěžovali, že vedle sebe nemají podobně velké kamarády, se kterými by si mohli povídat.

Tak se Kevin rozhodl, že je v klíčkách seřadí podle velikosti. Pro pořádek vždy vyndá jen dva skřítky a vymění jejich pozice. Navrhnete algoritmus, který má Kevin použít, aby skřítky seřadil podle velikosti od nejmenšího po největšího na co nejméně prohození.

**Příklad:** Pokud máme tři skřítky zavřené v klíčkách v pořadí: 3, 1, 2 (čím větší číslo, tím větší skřítek), tak je můžeme seřadit na dvě prohození. Nejprve prohodíme skřítku 3 se skřítkem 1 a poté skřítky 3 a 2.



Chcete-li s námi komunikovat šifrovaně a bezpečně, můžete si ověřit náš HTTPS certifikát – jeho SHA1 fingerprint je: **0E:D9:B6:E5:6F:B0:51:D9:66:EB:E9:29:E4:58:AB:5F:99:D6:FD:A3**.

# Korespondenční Seminář z Programování

## ZAČÁTEČNICKÁ KATEGORIE

26. ročník

KSP-Z

Listopad 2013

Právě se vám do rukou dostává historicky první zadání začátečnické kategorie KSP. Ta by oproti klasickému KSP měla být mnohem snazší a přístupná třeba i lidem, kteří se doposud s programováním a informatikou potkali jen velmi letmo.

KSP-Z se snaží být odrazovým můstkem pro každého, kdo se chce naučit používat počítač nejen ke čtení emailů a je zvědavý na taje informatiky.

Budete v něm řešit úlohy jak ryze praktické, kde vám budete posílat jen výsledky na zadané vstupy, tak úlohy teoretické, kde bude naopak záležet hlavně na návrhu a popisu algoritmu. Výsledek praktických úloh zjistíte ihned, teoretické úlohy vám opravíme po skončení série. Ke všem úlohám zveřejníme také vzorová řešení, ze kterých máte možnost se poučit. Pak na jaře ty nejúspěšnější z vás pozveme na týdenní soustředění v přírodě, kde své schopnosti ještě zdokonalíte a zároveň si užijete spoustu zábavy.

Termín odevzdání první série je stanoven na **pondělí 6. ledna 2014 v 8:00 SEČ**.

Řešení přijímáme elektronicky na <https://ksp.mff.cuni.cz/z/>. Tam také můžete nalézt další informace o tom, jak KSP a KSP-Z fungují. Rovněž tam najdete fórum, kde se můžete na cokoli zeptat. Anebo nám můžete napsat na email [ksp@mff.cuni.cz](mailto:ksp@mff.cuni.cz).



### Zadání první série začátečnické kategorie 26. ročníku KSP

První čtyři úlohy (označené piktogramem počítače) jsou praktické. To znamená, že ke každé z nich máme na webu ke stáhnutí několik sad vstupů a body získáš za odevzdání správných výstupů k odpovídajícím sadám. Jak výstupy vygeneruješ (třeba v jakém jazyce napíšeš program, který je zpracuje), je už čistě na Tobě.

#### 26-Z1-1 Kevin a magnety 8 bodů

Kevin má v krabičce  $N$  magnetů – obyčejných magnetických tyček, které mají na jednom konci pól  $+$  a na druhém  $-$ . Jednoho dne se rozhodl, že si v nich udělá pořádek.

Jeden po druhém magnety vyndával z krabičky a pokládal je do řady na stůl. Někdy se spojily k sobě (pokud se setkaly opačnými póly), jindy se mezi nimi zase vytvořila mezera.

Kevinu by zajímalo, do kolika částí se mu magnety spojily.

**Tvar vstupu:** Program na vstupu dostane na prvním řádku číslo  $N$  ( $1 \leq N \leq 1000000$ ). Každý z dalších  $N$  řádků bude obsahovat buď řetězec  $+-$ , nebo  $-+$  udávající, kterým směrem Kevin magnet položil.

**Tvar výstupu:** Na výstup vypíše jediný řádek obsahující jedno celé číslo: počet oddělených částí, do kterých se magnety spojí.

**Ukázkový vstup:** **Ukázkový výstup:**

```
4
+-
+-
--
--
```

#### 26-Z1-2 Piškvorky 10 bodů

Kevin sedí na hodině dějepisu. Bitvy, letopočty, děsná nuda. Raději si hraje se svým novým smartphonem a píše si s kamarádkou Sárrou, která se ve vedlejší třídě nudí na hodině matiky. Aby si zkrátili dlouhou chvíli, rozhodli se hrát piškvorky.

Jejich aplikace na piškvorky ale není dokonalá a neumí poznat, jestli už někdo vyhrál. To si Kevin se Sárrou uvědomili

až po nějaké době, a tak by je nyní zajímalo, kolikrát kdo z nich vyhrál. Kevin má křížky a Sára kolečka.

**Tvar vstupu:** Na vstupu dostanete čísla  $S$  a  $R$  udávající šířku a výšku hrací plochy ( $1 \leq R, S \leq 1000$ ). Na dalších  $R$  řádcích bude vždy  $S$  znaků  $X$ ,  $O$ , nebo  $.$ , kde  $X$  znamená křížek,  $O$  kolečko a  $.$  znamená prázdné políčko hrací plochy.

**Tvar výstupu:** Na výstup vypíše jediný řádek se dvěma celými čísly oddělenými mezerou: počtem pětic křížků a počtem pětic koleček. Pětice mohou ležet v libovolném řádku, sloupci, nebo úhlopříčce.

**Ukázkový vstup:** **Ukázkový výstup:**

```
6 6
XXXXXO
.X. .OX
..XO.X
..OX.X
.O..XX
00000X
```

#### 26-Z1-3 Zamilovaný dopis 10 bodů

Kevin působí jako bubeník ve studentské rockové kapelě *Velká tlama*. Kapela během svého prvního veřejného koncertu sklídila ohromný úspěch. Další den Kevin objevil ve schránce  $K$  dopisů. Mezi všemi nudnými úředními dopisy jeden vyčníval. Byl navoněný a na obálce měl nakreslené růžové srdíčko. Ten si okamžitě přečetl. Byl od fanynky!



Takové dopisy Kevin nedostává každý den. Obsah dopisu si tak hned slovo od slova zapamatoval. Bohužel pak při cestě

do školy přešlo a dopisy Kevinovi zmokly. Některé se staly téměř nečitelnými. I tak se ale snažil a přečetl z nich, co jen šlo. Posloupností těchto znaků si zapsal. Nyní by chtěl vědět, který z těchto dopisů by mohl být oním dopisem se srdíčkem.

**Tvar vstupu:** Na prvním řádku vstupního souboru bude zadán text původního dopisu, jehož délka bude nejvýše 300 000 znaků. Na druhém řádku bude zadáno číslo  $K$  udávající celkový počet doručených dopisů ( $1 \leq K \leq 50$ ). Na každém ze zbylých  $K$  řádků bude posloupnost maximálně 300 000 znaků, které se Kevinovi z daného dopisu podařilo přečíst. Všechny znaky dopisů budou složeny z malých a velkých písmen anglické abecedy a znaků  $.$ ,  $,$ ,  $_$ , kde  $_$  odděluje jednotlivá slova.

**Tvar výstupu:** Vypíše  $K$  řádků výstupu. Na  $i$ -tý řádek umístíte řetězec ANO, pokud  $i$ -tý dopis může být oním dopisem se srdíčkem. To znamená, že  $i$ -tý dopis lze z původního dopisu vytvořit vynecháním některých znaků. V opačném případě vypíše řetězec NE.

**Ukázkový vstup:** **Ukázkový výstup:**

```
Horoucne_Te_miluji,_Kevine. NE
5 ANO
Horoucne_Te_milujeme,_Kevine. NE
nemiluji ANO
MILUJI. ANO
Hor___eve.
Horce_Te_miluji,Kene.
```

#### 26-Z1-4 Hroch v jezeře 12 bodů

Kevin po odpoledních hraje počítačovou hru Hroch v jezeře. Ve hře plave s hrochem ve velkém jezeře, ze kterého se tu a tam vynořuje ostrůvek. Některé ostrůvky jsou pusté, na jiných se nachází hroší laskominy – jídlo všeho druhu. Cílem hry je doplavit s hrochem na ostrůvek s co největším množstvím jídla a mezitím nevylézat z vody.

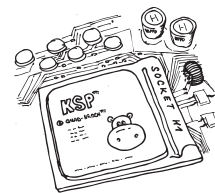
Ostrov je souvislá oblast políček pevniny, kde políčka považujeme za spojená, pokud se dotýkají hranou (nikoliv pouze rohem). Hroch může plavat nahoru, dolů, doleva a doprava.

**Tvar vstupu:** Na prvním řádku vstupu dostanete čísla  $S$  a  $R$  udávající šířku a výšku mapy hry. Na každém z dalších  $R$  řádků bude  $S$  znaků, přičemž:

- $.$  znamená, že na políčku je voda,
- $\#$  znamená, že na políčku je pevnina bez jídla,
- $J$  je pevnina s jídlem,
- $H$  je políčko, kde hroch začíná.

Zaručujeme, že na vstupu bude právě jedno políčko se znakem  $H$ .

**Tvar výstupu:** Na výstup napíše jedině celé číslo udávající největší množství jídla na ostrově, ke kterému se hroch může ze své startovní pozice doplavit povolenými způsoby bez toho, aby mezitím vystoupil z vody.



**Ukázkový vstup:**

```
10 8
.J##..#.J#
..###.J#.J
#JJJ#..#.J
..###..J.J
.....#.J
.J.H.###.J
.J...#JJ.#
.J#..#JJ.J
```

**Ukázkový výstup:**

6

Zbylé dvě úlohy této série jsou teoretické. K těmto úlohám je třeba vymyslet algoritmus a slovně ho popsat – tedy primární část řešení není zdrojový kód v nějakém programovacím jazyce, ale slovní popis. Sepsanou úlohu odevzdejte opět na našem webu.

#### 26-Z1-5 Úkol z geometrie 12 bodů

Kevin a jeho spolužáci dostali na hodině geometrie následující úkol. Mají zadaných  $N$  bodů na ose  $x$ . Je třeba každé dva po sobě jdoucí body spojit horní půlkružnicí. Kevin se podíval na posloupnost ze zadání a začal přemýšlet, jestli se některé dvě půlkružnice protnou. Navrhnete co nejefektivnější algoritmus, který pro zadanou posloupnost  $N$  bodů odpoví ANO, pokud se některé půlkružnice protínají, a NE, pokud nikoliv.

Na prvním obrázku je příklad pro posloupnost 0, 8, 20, 14, 18 a na druhém pro posloupnost 0, 12, 4, 10, 20. V prvním se kružnice neprotínají, v tom druhém protínají.



#### 26-Z1-6 Nezbední skřítkci 14 bodů

Kevin byl na vánočních trzích a co nesehnal – opravdové, živé vánoční skřítky! Tomu nemohl odolat a hned si jich  $N$  koupil. Dokonce byl každý jinak velký.

Aby mu neutekli, vyrobil si pro ně doma řadu  $N$  klíček a každého do jedné zavleč. Zavření skřítkci ale dělali příšerný kravál a neustále si stěžovali, že vedle sebe nemají podobně velké kamarády, se kterými by si mohli povídat.

Tak se Kevin rozhodl, že je v klíčkách seřadí podle velikosti. Pro pořádek vždy vyndá jen dva skřítky a vymění jejich pozice. Navrhnete algoritmus, který má Kevin použít, aby skřítky seřadil podle velikosti od nejmenšího po největšího na co nejméně prohození.

**Příklad:** Pokud máme tři skřítky zavřené v klíčkách v pořadí: 3, 1, 2 (čím větší číslo, tím větší skřítek), tak je můžeme seřadit na dvě prohození. Nejprve prohodíme skřítku 3 se skřítkem 1 a poté skřítky 3 a 2.



Chcete-li s námi komunikovat šifrovaně a bezpečně, můžete si ověřit náš HTTPS certifikát – jeho SHA1 fingerprint je: **0E:D9:B6:E5:6F:B0:51:D9:66:EB:E9:29:E4:58:AB:5F:99:D6:FD:A3**.