

PROG1: cvicne priklady
Tyzdne 1-8

Sekcia 1: Podmienene prikazy

1. Definujte funkciu, s nasledovnymi vlastnostami. Funkcia bude mat styri parametre: cisla a, b, c, d . Funkcia vrati hodnotu True, ak je interval (a, b) podmnozinou intervalu (c, d) , alebo ak je interval (c, d) podmnozinou intervalu (a, b) . Inak vrati funkcia hodnotu False. Mozete predpokladat, ze $a < b$ a $c < d$.

2. Definujte funkciu, s nasledovnymi vlastnostami. Funkcia bude mat styri parametre: cisla a, b, c, d . Funkcia vrati hodnotu True, ak je prienik intervalov (a, b) a (c, d) neprazdny. Inak vrati funkcia hodnotu False. Mozete predpokladat, ze $a < b$ a $c < d$.

3. Definujte funkciu, s nasledovnymi vlastnostami. Funkcia bude mat tri parametre: cisla a, b, x . Funkcia vrati cislo a , ak je cislo x blizsie k cislu a ako k cislu b . Inak funkcia vrati cislo b . Mozete predpokladat, ze cislo a je mensie ako cislo b .

Sekcia 2: Retazce

1. Definujte funkciu F s nasledujucimi vlastnostami. F ma dva parametre: retazec $ret1$ a retazec $ret2$. F vrati hodnotu True, ak sa niektery znak z retazca $ret1$ nachadza v retazci $ret2$. V opacnom pripade vrati hodnotu False.

2. Definujte funkciu F s nasledujucimi vlastnostami. F ma dva parametre: retazec $ret1$ a retazec $ret2$. F vrati hodnotu True, ak sa kazdy znak z retazca $ret1$ nachadza v retazci $ret2$. V opacnom pripade vrati F hodnotu False.

3. Definujte funkciu F s nasledujucimi vlastnostami. F ma jeden parameter: retazec ret . F vrati hodnotu True, ak sa niektery znak v retazci ret opakuje. V opacnom pripade vrati F hodnotu False. Ulohu rieste bez pouzitia metody count.

4. Definujte funkciu F s nasledovnymi vlastnostami. F ma jeden parameter: retazec ret . Predpokladajte, ze retazec ret je neprazdny, ze obsahuje iba male pismena abecedy a ze ziadne pismo sa v nom neopakuje. Funkcia vrati to pismo z retazca ret , ktore sa v abecede vyskytuje najneskor. Priklad: pre retazec 'otec' funkcia vrati 't'.

Sekcia 3: Rekurzia

1. Definujte funkciu s parametrom n . Funkcia nacita z klavesnice n retazcov a vrati najdlhsi z nacistanych retazcov. Vo funkcii pouzite **rekurziu**! Ak ma viacero retazcov maximalnu dlzku, funkcia vrati ten retazec s maximalnou dlzkou, ktory bol nacistany ako prvý.

2. Definujte funkciu F s nasledujúcimi vlastnostami. F má jeden parameter: retazec ret . F vráti počet velkych pismen v retazci ret . Vo funkcii pouzite metodu `isupper()`. Ziadnu inu metodu pouzít nemozete. Funkciu definujte **rekurzivne**!

Pomocka: Pre lubovolny retazec ret vrati metoda `isupper()` hodnotu `True`, ak plati, ze vsetky pismena v retazci ret su velke a zaroven plati, ze retazec ret obsahuje aspon jedno pismeno. Inak vrati metoda `isupper()` hodnotu `False`.

3. Definujte funkciu s nasledovnymi vlastnostami. Funkcia bude mat dva parametre: retazec ret a znak x . Funkcia vrati hodnotu `True`, ak sa znak x vyskytuje v retazci ret . Inak vrati funkcia hodnotu `False`. Ulohu vyrieste bez pouzitia operatora `in` a metody `find`. Vo funkcii pouzite **rekurziu**!

4. Definujte funkciu s nasledovnymi vlastnostami. Funkcia bude mat dva parametre: retazec ret a znak x . Funkcia vrati pocet vyskytov znaku x v retazci ret . Vo funkcii pouzite **rekurziu**! Ulohu rieste bez pouzitia metody `count`.

5. Definujte funkciu s parametrom n . Funkcia nacita z klavesnice n retazcov a vrati pocet, kolko z nacistanych retazcov malo dlzku vacsiu ako 5. Vo funkcii pouzite **rekurziu**!

6. Definujte funkciu s parametrom n , ktora vrati hodnotu vyrazu $1-2+3-4+\dots+(-1)^{n+1}n$. Vsimnite si, ze tento vyraz obsahuje vsetky cisla od 1 po n a ze neparne cisla vystupuju v tomto vyraze so znamienkom $+$ a parne cisla so znamienkom $-$. Vo funkcii pouzite **rekurziu**!

7. Definujte funkciu s nasledovnymi vlastnostami. Funkcia bude mat jeden argument: retazec ret . Funkcia vrati hodnotu `True`, ak retazec ret je palindrom. Inak vrati funkcia hodnotu `False`. Vo funkcii pouzite **rekurziu**.

Poznamka: Palindrom je retazec, ktory sa cita rovnako odpredu aj odzadu (napríklad 'anna', 'oko').

Sekcia 4: While cykly

Poznámka: Ulohy v tejto sekcii rieste bez použitia zoznamov!

1. Definujte funkciu s nasledujucimi vlastnostami. Funkcia bude mat parameter k . Funkcia bude nacitavat cisla z klavesnice, kym sa nestane, ze sa k -krat za sebou nacitalo rovnake cislo. Funkcia vrati hodnotu posledneho nacitaneho cisla.

Priklad: Ak $k=3$ potom funkcia prestane nacitavat cisla, ak sa napriklad zadali cisla 1,2,2,1,2,2,2 a vrati cislo 2.

2. Definujte funkciu s nasledujucimi vlastnostami. Funkcia bude nacitavat cisla klavesnice, kym pouzivatel nezada 0. Funkcia vrati pocet rovin v postupnosti nacitanych cisel.

Vysvetlenie: Ak mame postupnost cisel, potom rovinou v tejto postupnosti cisel budeme nazývat podpostupnost po sebe iducich rovnakych cisel s dlzkou minimalne 2. (Priklad: v postupnosti {7,7,7,6,6,2,3,2,6,6,6,6,0} sa nachadzaju tieto tri roviny: {7,7,7}, {6,6}, {6,6,6,6})

3. Definujte funkciu, ktora bude nacitavat cisla z klavesnice, kym uzivatel nezada nulu. Po zadani nuly funkcia vrati pocet kolkokrat sa stalo, ze nacitane cislo bolo vacsie ako sucet 2 cisel nacitanych pred nim. Napriklad pre postupnost cisel 1,7,9,2,4,-1,-2,0 funkcia vrati cislo 2, pretoze $9 > 7+1$ a $0 > (-1)+(-2)$.

4. Definujte funkciu s nasledujucimi vlastnostami. Funkcia bude nacitavat cisla z klavesnice, kym pouzivatel nezada 0. Funkcia vrati pocet lokalnych maxim v postupnosti nacitanych cisel.

Vysvetlenie: Ak mame postupnost cisel, potom cislo x z tejto postupnosti budeme nazývat lokalnym maximom, ak cislo, ktore sa v postupnosti nachadza pred x , je mensie ako x , a aj cislo, ktore sa v postupnosti nachadza za x , je mensie ako x . (Priklad: v postupnosti {7,5,6,4,2,3,0} su lokalnymi maximami cisla 6 a 3.)

5. Definujte funkciu s nasledujucimi vlastnostami. Funkcia bude nacitavat retazce z klavesnice, kym sa nestane, ze funkcia dvakrat po sebe nacita retazec zacinajuci na rovnaky znak. Funkcia vrati pocet nacitanych retazcov. Mozete predpokladat, ze vsetky nacitane retazce su neprazdne.