

STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

MNOŽSTVÍ INFORMACE VE ZPRÁVĚ, REPREZENTACE ČÍSEL A ZNAKŮ V POČÍTAČI

Zapište binární kód reálného čísla 35,125 v systému kódování s plovoucí desetinnou čárkou IEEE 754. využijte formát binary16. Jeho rozsahy jsou uvedeny v tabulce níže.

	Znaménko	Exponent	Mantisa
Velikost	1b	5b	11b

STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

DIGITALIZACE AKUSTICKÉHO SIGNÁLU, RASTROVÁ REPREZENTACE OBRAZU

Rastrový obrázek má rozlišení 4000 x 2000 pixelů. U obrázku není použita žádná forma komprese. Sami si zvolte co nejmenší barevnou hloubku, obrázek musí být ale schopen zobrazit alespoň 12 různých barev. Jaká bude teoretická minimální velikost příslušného souboru v megabajtech? U výsledku uveďte správnou zkratku jednotky.

STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

METODY KOMPRESY DAT

Máme zprávu aaabaacaaaba. Známe tabulku předpokládaných pravděpodobností jednotlivých znaků:

Znak	a	b	c	d	e
Pravděpodobnost	0,6	0,25	0,125	0,1	0,025

1. Provedte kompresi této zprávy pomocí LZW komprese

Spočítejte kompresní poměr, pokud víme, že velikost původní zprávy byla 36 b.

STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

RELACE

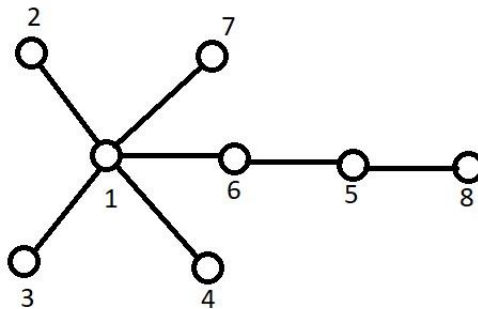
Pět přátel, Petr, Jan, Lukáš, Pavel a Jiří se nemohou dohodnout, kdo bude kapitánem jejich družstva. Nakonec se rozhodnou, že budou hrát hru Kámen-Nůžky-Papír. Spočítejte:

- a) Kolik různých variant má jedno kolo hry (tj. jedno stříhnutí).
- b) Jaká je pravděpodobnost, že hra neskončí po prvním kole.
- c) V této hře je relace „A poráží B“ (např. kámen poráží nůžky). Napište, o jaký typ relace se jedná.

STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

GRAFY

Prüferův kód se využívá na zakódování grafu, který má strukturu stromu. Strom se využívá v mnoha dalších aplikacích, například pro kompresi (Huffmanovo kódování) apod. Prüferův kód umožňuje efektivní reprezentaci stromu. Na obrázku máte zadaný strom, zakódujte ho pomocí Prüferova kódu.



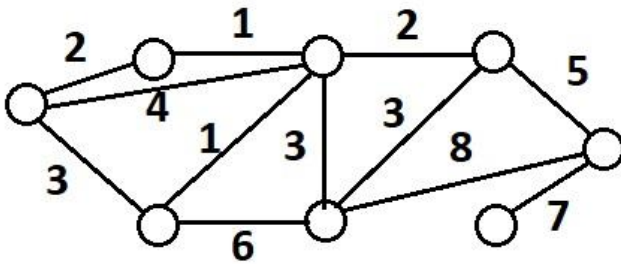
STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

PRINCIPY VYBRANÝCH GRAFOVÝCH ALGORITMŮ

Následující graf znázorňuje možná propojení různých počítačových zařízení (vrcholy) pomocí kabelů (hrany).

a) Najděte a nakreslete řešení, kdy propojíte všechna zařízení, a přitom spotřebujete minimální délku kabelů. Záložní propojení neuvažujeme. Popište váš algoritmus hledání řešení strukturovaným popisem nebo pomocí pseudokódu.

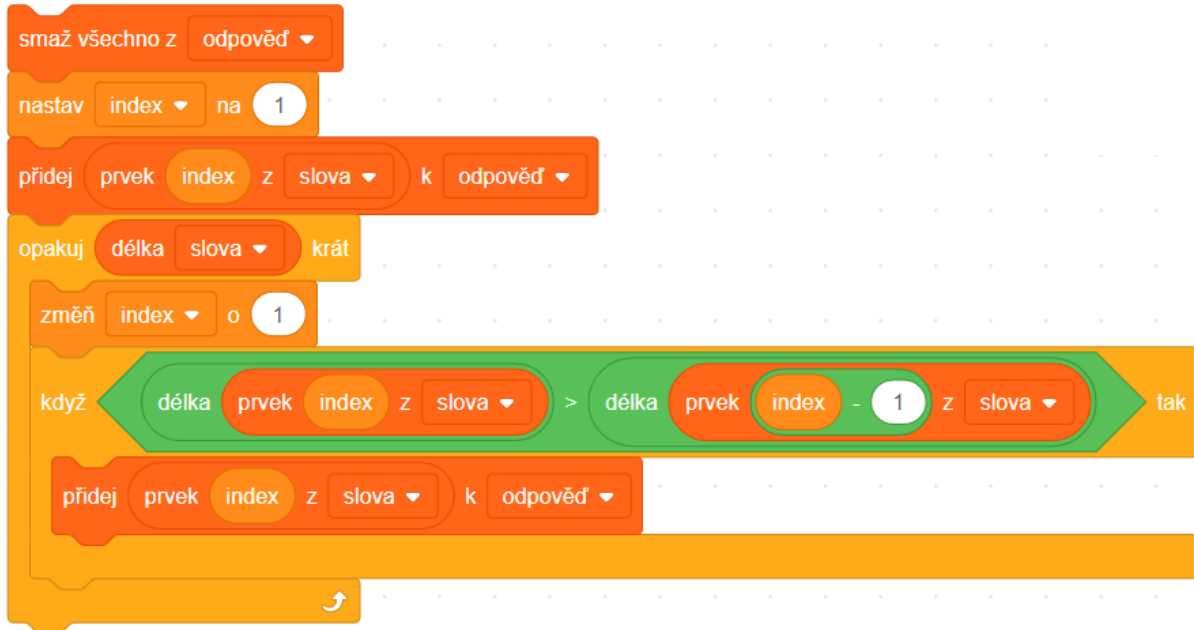
b) V kterém případě nalezneme v grafu více takových řešení? Popište také případy a demonstруйте to na zadaném grafu.



STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

BLOKOVÉ PROGRAMOVÁNÍ

a) Vysvětlete, co udělá scénář na obrázku.



- b) Před spuštěním scénáře vypadá seznam *slova* jako na obrázku vpravo. Jak budou vypadat oba seznamy po vykonání scénáře?
- c) Obsahuje scénář chybu? Jestliže ano, jakou a jak ji opravíte?
- d) Přidejte ke scénáři další bloky tak, aby po jeho vykonání vypadal seznam *slova* stejně jako seznam *odpověď*.

slova	
1	já
2	mám
3	rád
4	nerad
5	koblíhy
6	rajskou
7	biftek
+ délka 7 =	

STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

BLOKOVÉ PROGRAMOVÁNÍ

scénář pro čtverec délka

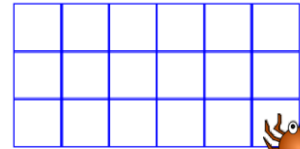
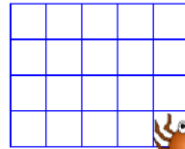
Je vytvořen , který nakreslí čtverec o délce strany dané parametrem a postava skončí ve stejné pozici, jako začala (horní obrázek).



Napište scénář pro nakreslení *okna*, složeného z těchto čtverců. Počet čtverců je dán dvěma parametry *vodorovně* a *visle*. Boční strana okna má mít celkovou délku 100 kroků.

Na obrázku vlevo je nakreslené okno pro parametry *vodorovně* = 5, *visle* = 4.

Na obrázku vpravo je nakreslené okno pro parametry *vodorovně* = 6, *visle* = 3.



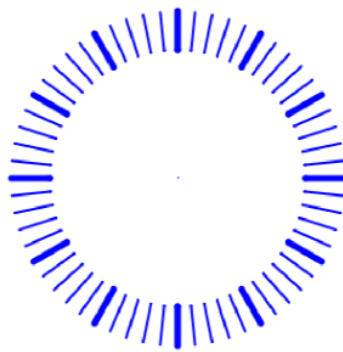
Scénář nemusíte zapisovat do bloků, stačí textově.

STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

BLOKOVÉ PROGRAMOVÁNÍ

- a) Napište scénář, který do ciferníku hodin nakreslí hodinové a minutové čárky. Scénář bude mít dva parametry: *od_středu* bude znamenat vzdálenost začátku čárky od středu ciferníku, *délka* bude znamenat délku čárky. Hodinové čárky budou mít větší tloušťku než minutové.

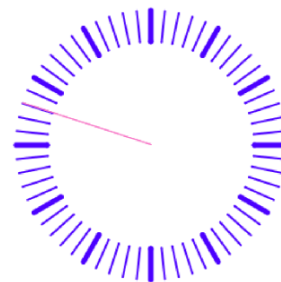
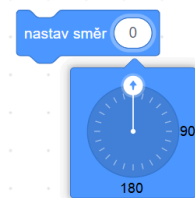
Není povoleno použít blok *nastav směr*.



- b) Sestavte scénář, který nakreslí minutovou ručičku, která bude ukazovat čas. Ten je uložen ve dvou proměnných *hodin* a *minut*. Minutová ručička bude dosahovat až na konec čárek.

Na obrázku vpravo ručička ukazuje 48 minut.

Můžete použít blok



STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

PROGRAMOVÁNÍ - PROGRAMOVÉ STRUKTURY

Níže uvedená metoda provádí výpočet ceny oběda na základě ceny polévky, hlavního jídla a zákusku. Přitom platí, že:

- polévka stojí vždy 40 Kč a informace, zda ji zákazník vyžaduje, je určena jedním z parametrů,
- cena hlavního jídla je proměnlivá a je určena dalším z parametrů,
- zákusek stojí vždy 50 Kč a informace, zda jej zákazník vyžaduje, je určena jiným z parametrů.

```
def spocitejCenuObeda(a, b, c):  
    vysledek = 0  
    if((b == True) and (c == True)):  
        vysledek = a + 40 + 50  
    elif(b == True):  
        vysledek = a + 40  
    elif(c == True):  
        vysledek = a + 50  
    else: vysledek = a  
    return vysledek
```

- Určete, pro jaké hodnoty parametrů bude metoda vracet hodnotu 300.
- Přejmenujte výstižně parametry a, b, c.
- Upravte kód tak, aby bylo možné do celkové ceny zahrnout i cenu předkrmu, který stojí vždy 30 Kč. Informace, zda jej zákazník vyžaduje, bude určena dalším vhodně pojmenovaným parametrem. Strukturu těla metody můžete zcela přepracovat.

STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

PROGRAMOVÁNÍ – DATOVÉ STRUKTURY

- a) Vysvětlete, co hledá následující metoda `hledej`. Předpokládejte, že v parametru `studenti` je seznam studentů a v parametru `cas` jsou časy, za jak dlouho tito studenti absolvovali určitý závod. Přitom platí, že hodnota na určité pozici v seznamu `cas` je čas studenta uvedeného na stejné pozici v seznamu `studenti`.

```
def hledej(studenti, casy):  
    x = casy[0]  
    y = studenti[0]  
    for i in range(0, len(studenti)):  
        a = len(studenti) - i - 1  
        if(x > casy[a]):  
            x = casy[a]  
            y = studenti[a]  
    print(y)
```

- b) Určete, co vypíše tato metoda, pokud bude využita v následujícím kódu:

```
s = ["Jirka", "Iva", "Lenka", "Petr", "Aleš", "Lukáš",  
    "Karolína"]  
c = [154, 134, 128, 147, 128, 164, 130]  
hledej(s, c)
```

- c) Uveďte, jak by provádění metody dopadlo, pokud by byly jako parametry `studenti` a `cas` použity prázdné seznamy. Svě tvrzení zdůvodněte.

STÁTNÍ ZÁVĚREČNÁ ZKOUŠKA – BC. ASISTENT SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ A
INFORMATIKU, UKÁZKOVÉ PROBLÉMOVĚ-PRAKTICKÉ ÚLOHY

OBJEKTOVÉ PROGRAMOVÁNÍ

Napište kód „konkrétní“ třídy **TestUloha**, která je odvozena od třídy **Uloha**. Doplňte:

- instanční proměnné **pocetOdpovedi** (int) , **odpovedi** (String[]) , **spravnaOdpoved** (int)
- **konstruktor** se 4 parametry (id, text, body a pocetOdpovedi) nastavující hodnoty odpovídajících instančních proměnných, přičemž inst. proměnnou **spravnaOdpoved** nastavte na -1
Je-li **pocetOdpovedi** menší než 2, dosadte do inst. proměnné **pocetOdpovedi** hodnotu 2
V konstruktoru alokujte pole **odpovedi** s počtem prvků odpovídajícím inst. proměnné **pocetOdpovedi**
- doplňte metodu nastavující správnou odpověď **setSpravnaOdpoved** (int spravna), která nastaví číslo správné odpovědi pouze tehdy, pokud je hodnota parametru v mezích indexů pole **odpovedi**

```
public abstract class Uloha {  
    private String id;  
    private final String text;  
    private int body;  
  
    public Uloha (String id, String text, int body) {  
        this.id = id;  
        this.text = text;  
        this.body = body;  
    }  
    ...  
}
```