

V následujícím textu jsou zadání tří úloh. Na splnění všech tří úloh budete mít celkem tři hodiny času. Je pouze na vás, kolik času budete věnovat které úloze a v jakém pořadí budete úlohy řešit.

Každá odevzdaná úloha bude hodnocena zvlášť a můžete za ní získat nejvýše tolik bodů, kolik je uvedeno u jejího názvu v hranatých závorkách. Za všechny tři úlohy je dohromady možné získat nejvýše 100 bodů. Úlohy se hodnotí ve čtyřech kategoriích: funkčnost, efektivita, uživatelské rozhraní a styl. U každé úlohy je napsáno, kolik bodů můžete za kterou kategorii získat. Jednotlivé kategorie jsou podrobně vysvětleny níže.

Za kategorii *funkčnost* dostanete body, pokud váš program dělá to, co bylo předepsáno zadáním úlohy, a dělá to správně. Dejte si pozor na to, že body vám mohou být strženy i v případě, že program sice bude většinou fungovat správně, ale v některých výjimečných případech vrátí špatný výsledek, skončí kvůli chybě nebo vůbec neskončí.

V kategorii *efektivita* se ve zkratce hodnotí, jak rychle váš program běží. Neznamená to ovšem, že byste se museli pouštět do nízkoúrovňových optimalizací nebo že některé programovací jazyky by byly výrazně znevýhodněny. Pro účely této kategorie je vcelku jedno, jestli program poběží na daném vstupu půl vteřiny nebo dvě vteřiny. Problém ale už bude, pokud program poběží místo půl vteřiny třeba čtvrt hodiny.

Chcete-li získat body v kategorii *styl*, snažte se psát svůj program tak, aby bylo co nejjednodušší ho po vás přečíst. K tomuto účelu můžete používat komentáře, srozumitelné názvy proměnných a funkcí, přehledné odsazování kódu a další podobné techniky.

Pokud je u hodnocení úlohy uvedena kategorie *uživatelské rozhraní*, můžete dostat body za to, jak bude program komunikovat s uživatelem. Není-li napsáno jinak, budou body uděleny za uživatelské rozhraní, které je přehledné a snadno pochopitelné a umožní uživateli efektivně a snadno s programem pracovat.

Přejeme vám mnoho štěstí v soutěži.

## Úloha 1 – Kalkulačka odmocnin [15]

Vytvoř program, který ze zadaného čísla vrátí odmocninu. Program musí podporovat odmocniny do 5-tého řádu (páté odmocniny) z kladných i záporných čísel. Přesnost výsledku stačí na 5 desetinných míst (0,00001). Řád odmocniny může být pouze z oboru přirozených čísel (nezáporné celé číslo).

Pokud má výsledek nenulovou komplexní část, bude výsledek zobrazen jako součet reálné a imaginární části.

Pro výpočet odmocnin lze použít Newtonovu metodu:

$$X_{y+1} = \frac{(n-1) \cdot X_y}{n} + \frac{b}{n \cdot X_y^{(n-1)}},$$

kde  $n$  je řád mocniny,  $b$  je číslo pod odmocninou,  $X_{y+1}$  je hledané číslo.

Výpočet se opakuje tak dlouho, dokud není  $|X_y - X_{y+1}| < \text{požadovaná přesnost}$  výsledku. Počáteční hodnotu  $X_y$  je vhodné nastavit jako 1. Ve druhém a každém dalším kroku se za  $X_y$  dosadí  $X_{y+1}$  z předchozího výpočtu.

Poznámka k výpočtu odmocnin:

Liché odmocniny ze záporných čísel lze počítat v oboru reálných čísel a zobrazit pouze jeden záporný výsledek. Sudé odmocniny ze záporných čísel je nutné počítat v oboru komplexních čísel. Znalosti komplexních čísel se lze elegantně vyhnout pokud známe výsledek odmocniny z -1:

Lze vyjít z předpokladu, že

$$\sqrt{-1} = \pm i \quad (\text{jedná se o 2 řešení})$$

$$\sqrt[4]{-1} = \pm 0,70710 \pm 0,70710i. \quad (\text{jedná se o 4 řešení})$$

$$\text{Tudíž platí} \quad \sqrt[n]{-b} = \sqrt[n]{-1} \cdot \sqrt[n]{b}$$

Ukázkový výstup programu:

Zadej číslo:

-16

Zadej řád odmocniny:

4

Počítám 4. odmocninu z -16

Výsledek je  $\pm 1,41421 \pm 1,41421i$

Hodnocení: *Funkčnost*: 7 bodů, *Efektivita*: 3 body, *Styl*: 3 body, *Uživatelské rozhraní*: 2 body

## Úloha 2 – Forenzní analýza [35]

14. ledna byla napadena kryptoměnová burza Cryptopia (cryptopia.co.nz). Novozélanští policisté ze speciální jednotky pro boj s kyberzločinem potřebují vysledovat, kde se nachází ukradené bitcoiny.

Napiš program na analýzu transakcí z napadených adres, který vypíše strom všech adres do n-té úrovně, na které byly odeslány bitcoiny ze zadané adresy po určitém datu (adresy 1. úrovně jsou všechny na které odešly transakce z výchozí adresy, adresy 2. úrovně jsou všechny na které odešly transakce z adres 1. úrovně atd.).

Uživatel zadá výchozí adresu, datum a úroveň, do které se mají transakce prohledávat. Vygenerován bude soubor ve formátu CSV (comma separated values) do složky, ze které je program spouštěn. Formátování CSV bude následující:

Zadaná adresa

1. úroveň

adresa1, zůstatek na adrese1

adresa2, zůstatek na adrese2

2. úroveň

adresa3, zůstatek na adrese3

adresa4, zůstatek na adrese4

adresa5, zůstatek na adrese1

adresa6, zůstatek na adrese2

Program by měl rozpoznat neexistující adresu a vypsát chybu.

Co by se mohlo hodit k řešení:

Aby nebylo nutné stahovat celý blockchain a prohledávat ho, má totiž přes 200GB, použij k získání dat o transakcích webový blockchain explorer. Např.: [www.blockchain.info](http://www.blockchain.info), transakční historii adresy lze stáhnout ve formátu JSON

<https://blockchain.info/rawaddr/<bitcoin adresa>>

Něco o bitcoinu:

Bitcoin je digitální měna, neexistují tedy žádné mince a bankovky, vše se odehrává mezi počítači na blockchainu. Blockchain je forma databáze, kde se zaznamenávají

všechny platby (v bance by to byly bankovní transakce) a lze z něj tak poznat, kdo má kolik bitcoinů. Přístup k bitcoinům je zprostředkován aplikací tzv. peněženkou, která spravuje jednu nebo i více bitcoinových adres. V bance by tomu odpovídalo číslo účtu (peněženka ~ adresa ~ číslo bankovního účtu). Na blockchainu se zaznamenávají transakce mezi adresami (jako převody mezi účty) a na základě transakcí lze určit kolik bitcoinů je na které adrese. Blockchain je dostupný všem a každý se může podívat kolik je v které peněžence bitcoinů. Musí znát ale její adresu.

Záznam z blockchainu lze získat v tzv. formátu JSON, což je vlastně jen obyčejný textový soubor s definovaným formátováním obsahu. Jsou tam uvedeny všechny transakce, které se k této adrese vztahují, tedy jak „odchozí“, tak „příchozí“. Každá transakce zde má vstup („inputs“), což je vlastně adresa plátce a plátcem odeslaná částka a výstup („out“), což je adresa příjemce platby a příjemcem přijatá částka. Aby to bylo ještě zamotanější, může mít každá platba (transakce) víc příjemců nebo i plátců, tedy víc vstupů nebo výstupů a tak se například odeslaná částka plátce, vstupující do transakce, může rozdělit na dvě či více částek pro více příjemců. Součet vstupních částek bitcoinů samozřejmě musí odpovídat součtu částek výstupních, neboť bitcoiny při transakci ani nevznikají, ani nezanikají.

Naším cílem je hledat všechny adresy příjemců, hledáme tedy všechny adresy v částech transakcí „out“. Hledáme řádky typu:

```
"addr": "1DPB9cZTYbfDBaxapYcmRKunkAzpm4KXhg"
```

```
a
```

```
"value": 58273227912,
```

což je hodnota transakce. Tato hodnota je v satoshi (obdoba haléřů u bitcoinu) a na bitcoiny se převede vydělením  $10^8$ , tedy 1 BTC = 100000000 těchto dílů.

Dále nás zajímá čas transakce, např.

```
"Time": 1448456143,
```

což je tzv. unix timestamp neboli čas v milisekundách od 1.1.1970, který je výpočetní technice často užíván. Pokud neznáte funkci nebo metodu na převod tohoto údaje na srozumitelnější časový údaj, vězte, že rok 2015 je ohraničen časy 1420070400 (1.1.2015 00:00 v UTC) a 1451606399 (31.1.2015 23:59 v UTC).

Pokud by program neuměl přistupovat k webu, lze k datům přistupovat lokálně ze souborového systému. K dispozici je k tomu určena sada několika souborů JSON ve složce Uloha2 <https://github.com/DDMP6/2019/tree/master/mladez>.

Hodnocení: *Funkčnost*: 25 bodů, *Efektivita*: 5 bodů, *Styl*: 3 body, *Uživatelské rozhraní*: 2 body

## Úloha 3 – OCR [50]

V dnešní době jsou úlohy rozpoznávání obrazu denním chlebem programátorů v nejrůznějších oblastech jako autonomní vozidla nebo FaceID.

Vytvoř program, který rozpozná číslo na obrázku. Vstupem bude cesta ke složce s obrázky ve formátu PNG, program dávkově rozpozná čísla na všech obrázcích a standardním výstupem budou názvy obrázků a číslice 0-9 nebo X pokud program vyhodnotí, že na obrázku není číslo. Příklad výstupu:

Zadej cestu ke složce:

C:\Users\MyUser\Dokumenty\Uloha3\testovací\_data

Vyhodnocuji ...

35\_00000.PNG – 5

X\_00001.PNG – X

X\_00002.PNG – 2

K dispozici je sada trénovacích obrázků a testovacích obrázků ve složce Uloha3, <https://github.com/DDMP6/2019/tree/master/mladez>

Za úspěšné řešení úlohy se považuje úspěšnost rozpoznání 80% na testovacích obrázcích.

Trénovací obrázky lze využít za běhu programu, program nemusí být předtrénovaný.

Hodnocení: *Funkčnost*: 22 bodů, *Efektivita*: 22 bodů, *Styl*: 3 body, *Uživatelské rozhraní*: 3 body

---

## INFORMACE

*Pořadatel obvodního kola*: Dům dětí a mládeže Praha 6 ve spolupráci FEL ČVUT.

### Vyhlášení výsledků:

- postupující soutěžící bude vyrozuměn e-mailem od krajského organizátora soutěže (<http://sp.stv.cz>)
- výsledky budou k dispozici na internetu na adrese [www.ddmp6.cz](http://www.ddmp6.cz)

### Postupující:

- **nejlepší 1-2 z každé kategorie postupují do krajského kola.**
- Termín Krajského kola bude stanoven v rozpětí 25. - 27. dubna 2019

### Předání cen :

**Slavnostní vyhlášení a předání hodnotných cen šesti nejlepším z každé kategorie proběhne v prostorách Fakulty elektrotechnické ČVUT (duben 2019), ceny od FEL ČVUT předá představitel fakulty a hosté z Magistrátu hl. m. Prahy a MČ Prahy 6.**

**Pozvánku na slavnostní předání cen obdrží každý účastník.**

**Dotazy:** Tomáš Košťál, tel. 235 323 333, e-mail: [tomas.kostal@ddmp6.cz](mailto:tomas.kostal@ddmp6.cz)

---

Dům dětí a mládeže Praha 6, U Boroviček 650/5, 163 00 Praha