

Základy procedurálneho programovania 2 (LS 2016/2017)

Skúška – 2. 6. 2017

MENO A PRIEZVISKO (PALIČKOVÝM PÍSMOM):

Vypíšte sľub a podpíšte ho:

„Vyhlasujem na svoju česť, že som túto písomku vypracoval(a) samostatne.“

Podpis:

Inštrukcie

1. Všetko odložte z lavice.
2. Na lavicu si vyberte jeden preukaz s fotografiou!
(študentský preukaz, občiansky preukaz, cestovný pas)
3. Na lavicu si pripravte písacie potreby!
(najlepšie aspoň TRI perá)
4. Pred sebou máte obojstranne vytlačené zadanie skúšky,
do ktorého vyplňate odpovede, a ktoré po skončení odovzdáte.
Iné (vlastné) papiere nie sú povolené! Nepíšte za okraj.

Skúška obsahuje 7 úloh, za ktoré môžete získať najviac 40 bodov.
5. Počkajte na zahájenie skúšky.
Skúška trvá 120 minút.

Veľa úspechov!

A (max. 5b): Vypočítaj hodnotu reťazca.

Daný je reťazec znakov, ktorý obsahuje aritmetickú úlohu s prirodzenými číslami a operáciami sčítania (+) a odčítania (-). Napíšte funkciu **vypocitaj**, ktorá vypočíta jeho výslednú číselnú hodnotu.

Reťazec obsahuje len znaky číslíc 0 až 9 a znamienka + a -, neobsahuje medzery alebo iné znaky.

Napr. pre reťazec: 12+23-34+567 je výsledok 568, a pre reťazec 1+2+3-1-2-3+1+2+3 je výsledok 6.

Môžete použiť funkciu **substr** na určenie podreťazca.

Ukážka zápisu reťazca 12+23-34+567 ako pole znakov:

znak	1	2	+	2	3	-	3	4	+	5	6	7	
ASCII	49	50	43	50	51	45	51	52	43	53	54	55	0
	0	1	2	3	4	5	6	7	8	9	10	11	12

```
int vypocitaj(char *str)
{
```

```
// Vrať podreťazec str[start..end]
// napr. substr("1234567", 2, 3) vráti: "34"
char *substr(char *str, int start, int end)
{
    char *buf = (char*) malloc(end-start+2);
    int i;
    for (i = start; i <= end; i++)
        buf[i-start] = str[i];
    buf[i-start] = 0;
    return buf;
}
```

```
}
```

B (max. 5b): Vymenujte aspoň 5 operácií pre prácu so spájaným zoznamom a napíšte hlavičky funkcií.

Odpoveď:

C (max. 5b): Spájané zoznamy, odstráň duplikáty v usporiadanom zozname. Uvažujte implementáciu spájaného zoznamu využitím štruktúr **Zoznam** a **Prvok** z predchádzajúcej úlohy, napíšte funkciu **odstran_duplikaty**, ktorá z usporiadaného spájaného zoznamu odstráni duplikáty tak, že vo výslednom zozname zostane len jeden výskyt z každej hodnoty.

Napr. zo zoznamu: Alica -> Alica -> Cyril -> Dano -> Dano -> Dano vytvorí zoznam: Alica -> Cyril -> Dano.

```
void odstran_duplikaty(struct Zoznam *z)
{

}

}
```

D (max. 2b): Čo vypíše nasledovný program? **Odpoveď:**

```
#include <stdio.h>

int main()
{
    unsigned int a = 9, b = 3, c = 0;
    while (b != 0) {
        if ((b & 1) != 0) c = c + a;
        a = a << 1;
        b = b >> 1;
    }
    printf("%u\n", c);
    return 0;
}
```

E (max. 4b): Napíšte funkciu **alokuj3d**, ktorá dynamicky alokuje trojrozmerné pole s rozmermi A x B x C. Rozmery dostane ako vstupné argumenty, výsledné alokované pole vráti ako návratovú hodnotu.

Odpoveď:

F (max. 4b): Napíšte postupnosť krokov ako sa zo zdrojových kódov (súbory .c a .h) vytvorí spustiteľný kód (.exe súbor). Jednotlivé kroky stručne vysvetlite. (Môžete aj nakresliť obrázok ak chcete).

Odpoveď:

G (max. 15b): Birds, Beasts and Fishes. Tradičná britská hra Birds, Beasts and Fishes u nás známa pod názvom Obesenec je jednoduchou hrou, pri ktorej hráč hľadá neznáme slovo alebo text skúšaním jednotlivých písmeniek. Vašou úlohou je vyhodnotiť variantu tejto hry, ktorú hrajú v televíznej šou, pri ktorej hráč najskôr vytočí na kolese bodovú hodnotu a až potom tipuje písmenko. Hráč získa za každé nájdené písmenko príslušný počet bodov, ktoré si vytočil na kolese. Hráči do televíznej šou chodia opakovane, vašou úlohou je určiť mená top troch hráčov a ich bodové zisky – teda tých troch hráčov, ktorí získali čo najviac bodov spomedzi všetkých hráčov, ak získali viacerí ako traja rovnakí počet bodov, stačí vypísať nejakých troch. Môžete predpokladať, že hru hralo najviac 100 hráčov.

V televíznej šou sa odohralo viacero hier, údaje o konkrétnej hre sú v súbore **hra_cislo.txt**, teda napr. **hra_1.txt** pre prvú hru, **hra_2.txt** pre druhú hru, atď. Počet hier nie je vopred zadáný, súbory načítavajte až kým existujú.

V súbore pre jednu hru je na prvom riadku najskôr uvedené meno hráča, na druhom riadku hľadaný (neznámy) text, a na každom ďalšom riadku až do konca súboru je počet bodov (celé číslo) a hádané písmenko (znak malej anglickej abecedy).

Napr. súbor hra_1.txt	znamená, že hráč vytočil 400 a hádal písmeno e, našiel 3 krát, potom
hugo	vytočil 200 a skúšal písmeno a, našiel raz, potom vytočil 500 a skúšal i,
bicykel nie je auto	našiel dva krát. Súťaž sa potom skončila, nie je potrebné, aby dohádaval
400 e	celý text. Hráč hugo pri tom celkovo získal $400 \cdot 3 + 200 \cdot 1 + 500 \cdot 2 =$
200 a	2400 bodov.
500 i	

Vašou úlohou je napísať program, ktorý vhodne (!) využije možnosti programovacieho jazyka C a spracuje údaje zo všetkých záznamov hier, a na štandardný výstup vypíše mená a bodové zisky top troch hráčov, ktorí získali najviac bodov spomedzi všetkých hráčov, ktorí sa hry zúčastnili.

Vhodne využite globálne polia/štruktúry (2 body) a vhodne rozčleňte program na nasledovné funkcie:

- (5 bodov) načítanie a vyhodnotenie súboru s údajmi o jednej hre – vstupný argument je smerník na už otvorený súbor (FILE *f)
- (2 bod) pre konkrétne písmenko, zistenie počtu písmeniek v reťazci
- (3 body) určenie top hráča spomedzi zostávajúcich hráčov (môžete volať opakovane, aby ste postupne našli top troch hráčov)
- (3 body) main funkcia (otváranie súborov, načítanie vstupu a výpis výsledku: top troch hráčov)

Odpoveď napíšte na prázdny list.