

## Zápočtový test KIV/IDT

28. 4. 2023

Příjmení a jméno:

Osobní číslo:

Cvičení:

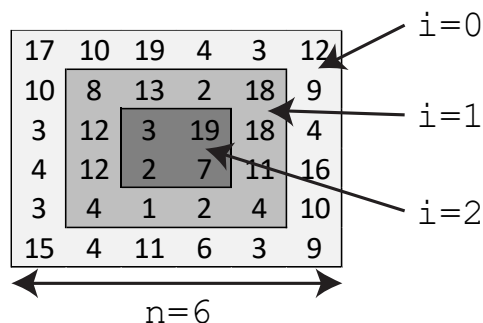
- ☐ Pondělí 11.10 (Frank)
- ☐ Pondělí 13.00 (Váša)
- ☐ Pondělí 14.50 (Frank)
- ☐ Úterý 11.10 (Potužák)
- ☐ Úterý 14.50 (Hotovec)
- ☐ Úterý 16.40 (Hotovec)
- ☐ Středa 8.25 (Potužák)
- ☐ Středa 13.00 (Hácha)
- ☐ Středa 14.50 (Hácha)
- ☐ Středa 16.40 (Hácha)
- ☐ Čtvrtek 12.05 (Majdišová)
- ☐ Čtvrtek 15.45 (Vítová)
- ☐ Pátek 9.20 (Majdišová)
- ☐ Pátek 9.20 (Vítová)

Body:

1	2	3	4	5	6	7	8	celkem

## Úloha 1 (6 bodů)

Napište metodu `int Onion(int[][] matrix, int n, int i)`, která ve čtvercové matici celých čísel velikosti  $n$  krát  $n$  reprezentované jako pole polí určí **součet** prvků v  $i$ -té slupce. Slupkou se myslí všechny prvky vzdálené od okraje právě  $i$  kroků (viz obrázek). Předpokládejte, že číslo  $n$  je **sudé** a všechny parametry jsou zadány **korektně** (kladná velikost matice, existující požadovaná slupka atd.).



## Úloha 2 (6 bodů)

Zásobník je reprezentován dynamickým polem. Pole bylo inicializováno na velikost 1, zvětšuje se vždy na **dvojnásobek**, používá se běžná strategie zvětšování (ze cvičení). Do zásobníku bylo postupně vloženo 17954 prvků, žádný prvek nebyl odebrán. **Kolikrát** došlo ke **zvětšení** pole? Kolik **prvků** pole bylo v rámci zvětšování **zkopírováno**? Odpovědi запиšte jako čísla nebo jako jednoznačně vyhodnotitelné výrazy.

### Úloha 3 (6 bodů)

Do následující neúplné implementace ADT Fronta doplňte metodu `Remove()` pro odebrání prvku.

```
class Link {
    public Link next;
    public int data;
}

class LinkQueue implements IQueue{
    Link first;
    Link last;

    public double Get() {
        if (first!=null)
            return first.data;
        else throw new Exception();
    }

}
```

### Úloha 4 (6 bodů)

Je dána funkce  $f(n) = 3n\sqrt{n} + 4n + 12,5$ . Určete, zda patří do následujících množin funkcí (množinu, do které funkce **patří**, označte křížkem):

- |                                      |                                      |  |   |
|--------------------------------------|--------------------------------------|--|---|
| <input type="checkbox"/> $O(1)$      | <input type="checkbox"/> $O(n)$      | <input type="checkbox"/> $O(n^2)$      | <input type="checkbox"/> $O(3n\sqrt{n} + 4n + 12,5)$      |
| <input type="checkbox"/> $\Omega(1)$ | <input type="checkbox"/> $\Omega(n)$ | <input type="checkbox"/> $\Omega(n^2)$ | <input type="checkbox"/> $\Omega(3n\sqrt{n} + 4n + 12,5)$ |
| <input type="checkbox"/> $\Theta(1)$ | <input type="checkbox"/> $\Theta(n)$ | <input type="checkbox"/> $\Theta(n^2)$ | <input type="checkbox"/> $\Theta(3n\sqrt{n} + 4n + 12,5)$ |

## Úloha 5 (7 bodů)

Následující program byl spuštěn bez parametrů příkazové řádky a zastavil se na vyznačené řádce, která dosud nebyla vykonána. Zapište celý obsah paměťové oblasti zásobník a halda. U všech pojmenovaných entit uveďte datový typ, jméno a hodnotu, u nepojmenovaných zapište třídu a všechny atributy (datové typy, názvy a hodnoty). Pokud je hodnotou reference, pak zakreslete šipku k instanci, na kterou reference odkazuje. Vyznačte zásobníkové rámce. Odpověď zapište na následující stranu.

```
class TriForce {
    TriForce power, wisdom, courage;

    void Attach(TriForce[] f, int i) {
        this.power = f[(i+1)%4];
        this.wisdom = f[(i+2)%4];
        this.courage = f[(i+3)%4];
        Console.WriteLine("Now");
    }

    public static void Main(String[] args) {
        TriForce[] forces = new TriForce[4];
        for (int i = 0; i < 4; i++)
            forces[i] = new TriForce();
        for (int i = 0; i < 4; i++)
            forces[i].Attach(forces, i);
    }
}
```



## Úloha 6 (7 bodů)

Je dán následující program:

```
int n = ...;
for (int i = 1; i < 100 * n; i++) {
    M(i, 0);
}
for (int j = 0; j < n; j++) {
    for (int i = 1; i < n; i++) {
        M(i, j);
        M(i, j);
        M(i, j);
        M(i, j);
    }
}
```

Zapište počet volání metody  $M(\dots)$  jako funkci proměnné  $n$ . Proved'te důkaz, že funkce patří/nepatří do množiny  $\Theta(n^2)$ .

## Úloha 7 (6 bodů)

Uvažte následující rekurzivní program:

```
static void Hanoi(char s, char t, char m, int c) {
    if (c == 0)
        return;
    else {
        hanoi(s, m, t, c-1);
        Console.WriteLine(s + "->" + t + ", ");
        hanoi(m, t, s, c-1);
    }
}
```

Následující program představuje neúplnou nerekurzivní variantu tohoto programu. Doplňte chybějící řádky.

```
class Task {
    public char s, t, m;
    public int c, segment;
    public Task(char s, char t, char m, int c) {
        _____;
        this.t = t;
        this.m = m;
        this.c = c;
        segment = 0;
    }
}

static void HanoiNR(char s, char t, char m, int c) {
    Stack<Task> tasks = new Stack<Task>();
    tasks.Push(new Task(s, t, m, c));
    while(tasks.Count>0) {
        Task task = tasks.Peek();
        switch (task.segment) {
            case 0:
                if (task.c == 0)
                {
                    tasks.Pop();
                    break;
                }
                _____;
                task.segment += 1;
                break;
            case 1:
                Console.WriteLine(task.s + "->" + task.t + ", ");
                tasks.Push(new Task(task.m, task.t, task.s, task.c-1));
                _____;
                break;
            case 2:
                tasks.Pop();
                break;
        }
    }
}
```

## Úloha 8 (6 bodů)

Mějme následující program:

```
class MyClassA {  
    virtual public void PrintSomething() {  
        Console.WriteLine("42");  
    }  
}  
  
class MyClassB : MyClassA{  
    override public void PrintSomething() {  
        Console.WriteLine("24");  
    }  
}  
  
class MyClassC : MyClassB{  
    override public void PrintSomething() {  
        Console.WriteLine("0");  
    }  
}
```

Co vypíše následující úsek programu?

```
MyClassA mc1 = new MyClassA();  
MyClassA mc2 = new MyClassB();  
MyClassA mc3 = new MyClassC();  
MyClassB mc4 = (MyClassB)mc2;  
mc1.PrintSomething();  
mc2.PrintSomething();  
mc3.PrintSomething();  
mc4.PrintSomething();  
Console.WriteLine(mc1 is MyClassA);  
Console.WriteLine(mc1 is MyClassB);  
Console.WriteLine(mc2 is MyClassA);  
Console.WriteLine(mc2 is MyClassB);  
Console.WriteLine(mc3 is MyClassA);  
Console.WriteLine(mc3 is MyClassB);  
Console.WriteLine(mc4 is MyClassA);  
Console.WriteLine(mc4 is MyClassB);
```