# 1. feladat

### Ekvivalencia partíciók

* 18 ≤ x ≤ 25 valid
* x < 18 invalid
* x > 25 invalid

### Tesztesetek a partíciókból vett véletlenszerű számokkal

Ezzel meg tudjuk erősíteni, hogy különbözőképpen viselkedik a rendszer a különböző partíciókból vett számokkal

T1

**Bemenet (input)**: 12,

**Elvárt eredmény(expected result)**: Kezelt kivétel/hiba (Handled exception/error)

T2

**Bemenet**: 20,

**Elvárt eredmény:** A rendszer elfogadja a beírt számot

T3

**Bemenet:** 39,

**Elvárt eredmény**: Kezelt kivétel/hiba

### Tesztesetek határérték elemzéssel

Az ekvivalencia partíciókból vett véletlenszerű számok nem lesznek alkalmasak arra, hogy bebizonyítsák azt, hogy pontosan a határértékek mentén változik a működés. Erre megoldás a határérték elemzés.

T1

**Bemenet**: 17,

**Elvárt eredmény**: Kezelt kivétel/hiba

T2

**Bemenet**: 18,

**Elvárt eredmény:** A rendszer elfogadja a beírt számot

T3

**Bemenet**: 25,

**Elvárt eredmény:** A rendszer elfogadja a beírt számot

**Bemenet**: 26,

**Elvárt eredmény**: Kezelt kivétel/hiba

Láthatjuk, hogy a határérték elemzéssel készített tesztesetek nagyobb lefedettséget adnak

* mivel a partíciók pontos határát vizsgálják, és
* logikailag tartalmazzák az ekvivalencia partíciók validálását is

### Extrém esetek:

Közvetlenül nem része a feladatnak, de megnézhetjük az extrém eseteket is

T4

**Bemenet**: “45kutya”,

**Elvárt eredmény**: Kezelt kivétel/hiba

T5

**Bemenet**: 17.9999999,

**Elvárt eredmény**: Kezelt kivétel/hiba

T6

**Bemenet**: “32”,

**Elvárt eredmény**: Kezelt kivétel/hiba (mivel stringként (szóként) adtam meg a bemenetet szám helyett)

# 2. feladat

### Ekvivalencia partíciók

* x < 24 - nincs meg a vizsga
* 24 ≤ x ≤ 40 - megvan a vizsga
* 40 < x - nincs információnk a helyes működésről

UPDATE : Hibás érték, mert nem érhetünk el 40+ pontot

### Tesztesetek határérték elemzéssel

T1

**Bemenet:** 23 pontot értünk el

**Elvárt eredmény**: nincs meg a vizsga

T2

**Bemenet:** 23 pontot értünk el

**Elvárt eredmény**: megvan a vizsga

T3

**Bemenet:** 40 pontot értünk el

**Elvárt eredmény**: megvan a vizsga

T4

**Bemenet:** 40 pontot értünk el

**Elvárt eredmény**: további pontosításra várunk mert a specifikációnk nem pontos

UPDATE : Kezelt hiba

# 8. feladat

A feltételek ebben a feladatban kétállásúak, szóval Igen/Nem, True/False, 1/0 értékekkel reprezentálhatjuk a választást. Ennek alapján a táblázat az alábbi módon tartalmazza az összes lehetséges kombinációt

| **Feltételek / Hatás** | **Szabály 1** | **Szabály 2** | **Szabály 3** | **Szabály 4** | **Szabály 5** | **Szabály 6** | **Szabály 7** | **Szabály 8** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **.jpg (Igen/Nem)** | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| **32kb (Igen/Nem)** | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| **137\*177 (Igen/Nem)** | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| **Eredmény - Hatás (Sikerült/Hiba)** | S | H | H | H | H | H | H | H |

Összesen 8 teszteset generálható

T1

**Bemenet:** 32kb-tól kisebb, az 137\*177 felbontástól kisebb jpg képet töltenek fel

**Elvárt eredmény**: Sikeres a feltöltés

T2

**Bemenet:** 32kb-tól kisebb, az 137\*177 felbontástól nagyobb jpg képet töltenek fel

**Elvárt eredmény**: Kezelt hiba

.

.

.

T8

**Bemenet:** 32kb-tól nagyobb, az 137\*177 felbontástól nagyobb png képet töltenek fel

**Elvárt eredmény**: Kezelt hiba

# 9. feladat

|  | szabály1 | szabály2 | szabály3 | szabály4 |
| --- | --- | --- | --- | --- |
| Bp/O | Bp | Bp | O | O |
| E/N | E | N | E | N |
| Eredmény | 1500 | 1000 | 2000 | 1500 |

T1

**Bemenet:** Budapest területére express szállítást kérünk

**Elvárt eredmény:** 1500

T2

**Bemenet:** Budapest területére normál szállítást kérünk

**Elvárt eredmény:** 1500

T3

**Bemenet:** Országos express szállítást kérünk

**Elvárt eredmény:** 2000

T4

**Bemenet:** Országos normál szállítást kérünk

**Elvárt eredmény:** 1500

# 14. feladat

Az állapotátmenet táblázat oszlop fejlécében felsoroljuk a lehetséges interakciókat.

Az első oszlopban pedig a lehetséges állapotokat. Az így meghatározott táblázat aktuális cellájába azt az állapotot írjuk, amelyikbe az adott kezdeti állapotból adott interakció hatására kerül. Így szisztematikusan bejárhatjuk az állapotgépünket, úgy hogy minden esetet lefedünk.

Start/1T a kezdő állapotunk

AG, AB a lehetséges végállapotaink

|  | **CP** | **IP** |
| --- | --- | --- |
| **Start / 1T** | AG | 2T |
| **2T** | AG | 3T |
| **3T** | AG | AB |
| **AG** |  |  |
| **AB** |  |  |

Ennek alapján mág könnyű lesz teszteseteket felvenni (6ot):

T1

**Bemenet:** Start/1T állapotban vagyunk, amikor helyes PINt adunk meg

**Elvárt eredmény:** A rendszer beenged minket

T2

**Bemenet:** Start/1T állapotban vagyunk, amikor helytelen pint adunk meg

**Elvárt eredmény:** a rendszert továbbra sem érjük el, de másodjára is megpróbálkozhatunk a PIN megadásával

.

.

.

T6

**Bemenet:** Harmadjára próbálkozunk a PIN megadásával, amikor harmadszorra sem jut eszünkbe

**Elvárt eredmény:** a rendszer kizár minket

# 15. feladat

|  | Pass | Fail |
| --- | --- | --- |
| 1A | H | 2A |
| 2A | H | 3A |
| 3A | H | AL |
| H |  |  |
| AL |  |  |

T1

**Bemenet:** 1A állapotban vagyunk, amikor helyes jelszót adunk meg

**Elvárt eredmény:** A rendszer beenged minket a home pagere

T2

**Bemenet:** 1A állapotban vagyunk, amikor helytelen jelszót adunk meg

**Elvárt eredmény:** a rendszert továbbra sem érjük el, de másodjára is megpróbálkozhatunk a megadásával

.

.

.

T6

**Bemenet:** Harmadjára próbálkozunk a jelszó megadásával, amikor harmadszorra sem jut eszünkbe

**Elvárt eredmény:** a fiókunkat zárolja rendszer