Kezdés ideje	2020. június 11., csütörtök, 13:16
Állapot	Befejezte
Befejezés dátuma	2020. június 11., csütörtök, 14:44
Felhasznált idő	1 óra 27 perc
Pont	**4** a maximum 5 közül (**80**%)

1 kérdés

Hibás

0 közül 1 leosztályozva

Készítsen ***v2m*** névvel függvényt, ami vektorból mátrixot készít. A függvény paraméterül várja a vektort, annak elemszámát és az előállítandó mátrix sorainak, oszlopainak számát. A függvényt úgy kell megvalósítani, hogy az a vektor elemeivel, a sorban az elsőtől az utolsóig haladva, feltölti a mátrixot. Először a mátrix első sorát töltse fel az első oszloptól az utolsóig haladva, majd a második sor következik, és így tovább a mátrix utolsó soráig. Feltételezheti, hogy a függvény mindig megfelelően kerül paraméterezésre, így nem kell azzal az esettel foglalkozni, ha a mátrix oszlop és sorszámának szorzata nem egyezik a vektor hosszával. A mátrixot tároló tömböt úgy alakítsa ki, hogy az első index a sort, a második az oszlopot határozza meg.

Answer: (penalty regime: 0 %)

Reset answer

```
1 double** v2m(double* v, int hossz, int sor, int oszlop) {}
```

Syntax Error(s)

```
__tester__.cpp: In function 'double** v2m(double*, int, int, int)':
__tester__.cpp:14:58: error: no return statement in function returning non-void [-
Werror=return-type]
```

```
 double** v2m(double* v, int hossz, int sor, int oszlop) {}
                                                          ^
cc1plus: all warnings being treated as errors
```

Question author's solution:

```
double** v2m(double* v, int hossz, int sor, int oszlop) {
  double** m = (double**)malloc(sor * sizeof(double*));
        for (int i = 0; i < sor; i++)
        {
                m[i] = (double*)malloc(oszlop * sizeof(double));
        }
  for(int i = 0; i < sor; i++) {
    for(int j = 0; j < oszlop; j++) {
      m[i][j] = v[i * oszlop + j];
    }
  }
  return m;
}
```

Hibás

Leadásra bejelöl: 0/1.

2 kérdés

Helyes

1 közül 1 leosztályozva

Készítsen ***darabszam*** függvényt, ami egy szivarok adatait tartalmazó listán megszámolja a Churchill és Toro méretű szivarokat.

A szivarok az alábbi struktúrában kerülnek tárolásra (a struktúrát nem kell létrehoznia):

```
struct szivar {
  double hossz;
  int atmero;
  szivar* kov;
};
```

Minden szivar esetén adott a hossza hüvelykben és a gyűrű mérete (átmérője).

A ***darabszam*** függvény bemenetként vár egy szivarokból álló láncolt listát, amin a megfelelő méretű szivarokat kell megszámolni. A darabszámokat a bemenetként kapott memóriaterületekre kell írni. Egy szivart akkor tekintünk Churchill méretűnek, ha hossza minimum 6,75 és maximum 8, valamint gyűrű mérete minimum 46 és maximum 48. Toro méretről akkor beszélünk, ha a szivar hossza minimum 5,5 és maximum 6,5, illetve a gyűrűje minimum 48 és maximum 54 méretű.

Answer: (penalty regime: 0 %)

Reset answer

```
1  void darabszam(szivar* szivarok, int* churchill, int* toro) {
2      *toro = 0;
3      *churchill = 0;
4
5      for (szivar* i = szivarok; i != NULL; i = i->kov) {
6          if ((i->hossz <= 6.5) and (i->hossz >= 5.5) and (i->atmero <= 54) and
7              *toro+=1;
8          }
9          if ((i->hossz <= 8) and (i->hossz >= 6.75) and (i->atmero <= 48) and
10             *churchill+=1;
11         }
12     }
13 }
14
```

	Test	Expected	Got	
✔	teszt();	OK	OK	✔

Passed all tests! ✔

Question author's solution:

```
void darabszam(szivar* szivarok, int* churchill, int* toro) {
  *churchill = 0;
  *toro = 0;
  while(szivarok != NULL) {
    if(szivarok->hossz >= 5.5 && szivarok->hossz <= 6.5 and szivarok->atmero >= 48 and
szivarok->atmero <= 54) {
      (*toro)++;
    }
    if(szivarok->hossz >= 6.75 && szivarok->hossz <= 8.0 and szivarok->atmero >= 46 and
szivarok->atmero <= 48) {
      (*churchill)++;
    }
    szivarok = szivarok->kov;
  }
}
```

Helyes

Leadásra bejelöl: 1/1.

3 kérdés

Helyes

1 közül 1 leosztályozva

Készítsen ***avg*** nevű függvényt, ami egy paraméterül kapott tömb elemeinek az átlagát veszi úgy, hogy a számítás során nem veszi figyelembe a tömb legnagyobb és a legkisebb elemét. Feltételezhető, hogy a tömbben egy érték csak egyszer szerepel, így nem kell azzal az esettel foglalkozni, ha több legkisebb vagy legnagyobb érték található, illetve azzal sem, ha a legkisebb és legnagyobb érték ugyanaz.

Answer: (penalty regime: 0 %)

Reset answer

```
1  double avg(double lista[], int hossz) {
2
3      double max=lista[0];
4      double min=lista[0];
5
6      for(int i=0; i<hossz; i++){
7          if(lista[i]>max) {
8              max = lista[i];}
9          if(lista[i]<min) {
10             min = lista[i];}
11     }
12
13     double ossz=0;
14     for (int j=0; j<hossz; j++){
15         ossz=ossz+lista[j];
16     }
17
18     double atlag = ossz - max - min;
19     atlag = atlag/(hossz-2);
20     return atlag;
21 }
22
```

	Test	Expected	Got	
✔	teszt();	OK	OK	✔

Passed all tests! ✔

Question author's solution:

```
double avg(double lista[], int hossz) {
  int min = 0;
  int max = 0;
  double avg = 0.0;
  for(int i = 1; i < hossz; i++) {
    if(lista[i] < lista[min]) {
      min = i;
    }
    if(lista[i] > lista[max]) {
      max = i;
    }
  }
  for(int i = 0; i < hossz; i++) {
    if(i != min && i != max) {
      avg += lista[i];
    }
  }
  avg /= hossz - 2;
  return avg;
}
```

Helyes

Leadásra bejelöl: 1/1.

4 kérdés

Helyes

1 közül 1 leosztályozva

Készítsen ***akkord*** nevű függvényt, ami megadja egy zenei hármashangzat hangjait az alaphang és a hangnem alapján.

A lehetséges zenei hangok sorrendben a következők (a programban nagybetűvel tárolva): C, C#, D, D#, E, F, F#, G, G#, A, B, H. A hangok folytonosan követik egymást, tehát a "H" hang után ismét "C" következik, majd "C#" és így tovább.

A függvény első paramétere az alaphang (nagybetűvel), majd a hangnem, ami "moll" vagy "dur" lehet (kisbetűvel, ékezet nélkül).

Ha a hangnem moll, akkor a hármashangzat sorrendben a következő hangokból áll: alaphang, alaphang után következő 3. hang, és az alaphang után következő 7. hang.

Ha a hangnem dúr, akkor a hármashangzat sorrendben a következő hangokból áll: alaphang, alaphang után következő 4. hang, és az alaphang után következő 7. hang.

A kimeneti string-ben egymástól egy-egy szóközzel elválasztva jelenjenek meg a hangok nevei.

Például:

Az alaphang C és a hangnem dúr, akkor a kimenet: "C E G".
Az alaphang A és a hangnem moll, akkor a kimenet: "A C E".

Feltételezheti, hogy a függvény helyesen kerül paraméterezésre, így a paraméterek helyességének vizsgálatával nem kell foglalkoznia.

Answer: (penalty regime: 0 %)

Reset answer

```
string akkord(string alap, string hangnem) {
    string hangok[]={"C", "C#", "D", "D#", "E", "F", "F#", "G", "G#", "A", "B
    string str="";
    if(hangnem=="dur"){
        int n=0;
        while(alap!=hangok[n]) n++;
        str = alap + " " + hangok[n+4] + " " + hangok[n+7];
    }
    if(hangnem=="moll"){
        int n=0;
        while(alap!=hangok[n]) n++;
        str = alap + " " + hangok[n+3] + " " + hangok[n+7];
    }
    return str;
}

```

	Test	Expected	Got	
✔	teszt();	OK	OK	✔

Passed all tests! ✔

Question author's solution:

```
string akkord(string alap, string hangnem) {
  string hangok[] = {"C", "C#", "D", "D#", "E", "F", "F#", "G", "G#", "A", "H", "B"};
  int i = 0;
  while(hangok[i] != alap) {
    i++;
  }
  string hangsor = hangok[i] + " ";
  if(hangnem == "moll") {
    hangsor = hangsor + hangok[(i + 3) % 12] + " " + hangok[(i + 7) % 12];
  } else if(hangnem == "dur") {
    hangsor = hangsor + hangok[(i + 4) % 12] + " " + hangok[(i + 7) % 12];
  }
  return hangsor;
}
```

Helyes

Leadásra bejelöl: 1/1.

5 kérdés

Helyes

1 közül 1 leosztályozva

Készítsen ***diff*** nevű függvényt, ami egy tömbben tárolt, időben változó digitális jelsorozat deriváltják közelíti véges differenciával. A tömb i-edik indexű eleme megadja, hogy a digitális jel milyen értéket vett fel az i-edik időpillanatban.

A véges differenciát az alábbi módon határozzuk meg: A kimeneti oldalon az i-edik időpillanathoz tartozó közelítő derivált érték legyen a bemeneti jel i-edik időpillanatban vett értékének és az (i-1)-edik időpillanatban vett értékének a különbsége. Ahol az (i-1)-edik időpillanat nem értelmezhető, ott a kimeneti oldalon 0 érték szerepeljen.

Answer: (penalty regime: 0 %)

Reset answer

```
double* diff(double jel[], int hossz) {
    double* derivalt = new double[hossz];
    derivalt[0] = 0;
    for (int i = 1; i < hossz; i++) {
        derivalt[i] = jel[i] - jel[i-1];
    }
    return derivalt;
}
```

	Test	Expected	Got	
✔	`teszt();`	OK	OK	✔

Passed all tests! ✔

Question author's solution:

```
double* diff(double jel[], int hossz) {
  double* kimenet = (double*)malloc(hossz * sizeof(double));
  kimenet[0] = 0;
  for(int i = 1; i < hossz; i++) {
    kimenet[i] = jel[i] - jel[i-1];
  }
  return kimenet;
}
```

Helyes

Leadásra bejelöl: 1/1.

◀ Mintavizsga Ugrás... Előadás fóliák ▶