

A1. feladat

Adott az alább definiált adattípus, amelyet síkbeli pontok adatainak tárolására használunk:

```
typedef struct {double x,y;} pont;
```

Írjon C függvényt, amely paraméterként egy pont típusú adatokból álló tömböt és annak elemszámát veszi át, majd a prímszámú elemeket az origó körül 90°-kal pozitív irányban elforgatja, míg a többi változatlanul hagyja!

```
1  typedef struct { double x, y; } point;
2
3  int is_prime(unsigned n)
4  {
5      int i;
6      for (i = 2; i*i <= n; ++i)
7          if ((n % i) == 0)
8              return 0;
9      return 1;
10 }
11
12 void rotate_prime(point points[], size_t n)
13 {
14     size_t i;
15     for (i = 0; i < n; ++i)
16     {
17         if (is_prime(i))
18         {
19             double x = points[i].x;
20             points[i].x = -points[i].y;
21             points[i].y = x;
22         }
23     }
24 }
```

A2. feladat

Egy pályázati kiírásra érkezett és már értékelt pályázatokról a pályázott összeget (maximum 20 000 000 Ft) és a pontszámot (0-100 pont) tartjuk nyilván. A pályázatoknak ezt a két fontos adatát egy tömbben tároljuk. Írjon C függvényt, amely egy ilyen tömböt kap, és visszaadja, hogy a pontozás alapján a legjobb 100 pályázatra mennyi pénzt kellene kifizetnie a pályázatotónak. Feltételezheti, hogy legkevesebb 100 pályázat beérkezett.

```
1 typedef struct { int pont; int osszeg; } palyazat;
2
3 int func(palyazat input[], int n) {
4     int i, sum;
5     palyazat csere;
6     for (i = 0; i < n; ++i) {
7         int j, iMax = i;
8         for (j = i; j < n; ++j)
9             if (input[j].pont > input[iMax].pont)
10                iMax = j;
11        csere = input[i];
12        input[i] = input[iMax];
13        input[iMax] = csere;
14    }
15
16    sum = 0;
17    for (i = 0; i < 100; ++i)
18        sum += input[i].osszeg;
19    return sum;
20 }
```

A3. feladat

Az `osztaly.txt` fájl minden sorában két adat van szóközzel elválasztva. Az első egy karakter (F vagy L), amely jelzi, hogy fiúról vagy lányról van-e szó. A második egy szó, az illető beceneve, amely legfeljebb 10 értékes karakterből áll. Írjon C programot, amely a `parok.txt` szöveges fájlba kiírja az összes, az osztály tanulóiból összeállítható heteroszexuális táncpárt. Minden sorba egy pár kerüljön, a pár első tagja a fiú, második tagja a lány legyen. A neveket szóközzel válassza el. Nem tudjuk, hogy hány fő van az osztályban, ezért használjon dinamikus adatszerkezetet. Az `osztaly.txt` fájlt csak egyszer olvashatja! Figyeljen rá, hogy memóriaszivárgás ne legyen a programban!

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5 typedef struct person {
6     char name[11];
7     struct person *next;
8 } person;
9
10 int main(void)
11 {
12     char gender;
13     char name[11];
14     person *boys = NULL, *girls = NULL, *pb, *pg;
15     FILE *fp = fopen("osztaly.txt", "r");
16
17     while (fscanf(fp, "%c%s\n", &gender, name) == 2) {
18         person *newper = (person *)malloc(sizeof(person));
19         strcpy(newper->name, name);
20         if (gender == 'F') {
21             newper->next = boys;
22             boys = newper;
23         }
24         else {
25             newper->next = girls;
26             girls = newper;
27         }
28     }
29     fclose(fp);
30
31     fp = fopen("parok.txt", "w");
32     for (pb = boys; pb != NULL; pb = pb->next)
33         for (pg = girls; pg != NULL; pg = pg->next)
34             fprintf(fp, "%s %s\n", pb->name, pg->name);
35     fclose(fp);
36
37     while (boys != NULL) {
38         pb = boys;
39         boys = boys->next;
40         free(pb);
41     }
42
43     while (girls != NULL) {
44         pg = girls;
45         girls = girls->next;
46         free(girls);
47     }
48
49     return 0;
50 }
```