1. U datoteci `math2.c` napisati definicije funkcija čiji su prototipovi prikazani u nastavku:

```
int binKoef(int m, int n);        // bin. koeficijent C(m, n) multiplikativnom formulom
double eksp(float x, int n);      // x na potenciju n, n ≥ 0
float fabsolut(float x);          // apsolutna vrijednost od x
double dabsolut(double x);        // apsolutna vrijednost od x
int iabsolut(int n);              // apsolutna vrijednost od n
unsigned long long
fakt(unsigned int n);             // n!
```

   Prototipove (deklaracije) navedenih funkcija smjestiti u datoteku zaglavlja `math2.h`

   Glavni program (funkciju `main`) smjestiti u datoteku `glavni.c`. U glavnom programu treba izračunati i na zaslon ispisati po jedan rezultat poziva svake od navedenih funkcija. Vrijednosti argumenata i format ispisa na zaslon odabrati po želji.

   Prevođenje programa testirati na dva načina:

   a) jednim pozivom prevodioca prevesti i povezati oba modula
   b) svaki od modula prevesti zasebno po jednim pozivom prevodioca, a zatim povezati dobiveni objektni kod

2. Proučiti sljedeći program. Čemu služe i kako rade funkcije `jestNeparan` i `jestParan`? Radi li se ovdje o nekoj vrsti rekurzije? Zašto se program, u obliku u kojem je trenutačno napisan, ne može prevesti? Bi li pomogla promjena redoslijeda funkcija u kodu? Što se mora dodati u program da bi se program mogao prevesti?

```
#include <stdio.h>

_Bool jestNeparan(unsigned int n) {
   _Bool jeNeparan = n == 0 ? 0 : jestParan(n - 1);
   return jeNeparan;
}

_Bool jestParan(unsigned int n) {
   _Bool jeParan = n == 0 ? 1 : jestNeparan(n - 1);
   return jeParan;
}

int main(void) {
   unsigned int n;
   scanf("%u", &n);
   if (jestParan(n))
      printf("%u je paran", n);
   else
      printf("%u je neparan", n);

   return 0;
}
```

Rješenja:

1. Datoteka **math2.h**

```
int binKoef(int m, int n);
double eksp(float x, int n);
float fabsolut(float x);
double dabsolut(double x);
int iabsolut(int n);
unsigned long long fakt(unsigned int n);
```

Datoteka **math2.c**

```
#include "math2.h"

int binKoef(int m, int n) {
   int rez = 1;
   int i;

   if (n < m - n)
      for (i = 1; i <= n; ++i)
         rez = rez * (m - n + i) / i;
   else
      for (i = 1; i <= m - n; ++i)
         rez = rez * (n + i) / i;

   return rez;
}

double eksp(float x, int n) {
   int i;
   double rez = 1.;
   for (i = 0; i < n; ++i)
      rez *= x;
   return rez;
}

float fabsolut(float x) {
   return x >= 0 ? x : -x;
}

double dabsolut(double x) {
   return x >= 0 ? x : -x;
}

int iabsolut(int n) {
   return n >= 0 ? n : -n;
}

unsigned long long
fakt(unsigned int n) {
   unsigned int i;
   unsigned long long umnozak = 1ULL;
   for (i = 2U; i <= n; ++i)
      umnozak = umnozak * i;
   return umnozak;
}
```

Datoteka **glavni.c**

```
#include <stdio.h>

#include "math2.h"

int main(void) {
   printf("binKoef(%d, %d) = %d\n", 12, 8, binKoef(12, 8));
   printf("eksp(%f, %d) = %f\n", 1.1f, -2, eksp(1.1f, 2));
   printf("fabsolut(%f) = %f\n", -5.1f, fabsolut(-5.1f));
   printf("dabsolut(%lf) = %lf\n", -5.1, dabsolut(-5.1));
   printf("iabsolut(%d) = %d\n", -5, iabsolut(-5));
   printf("fakt(%u) = %llu\n", 12U, fakt(12U));

   return 0;
}
```

```
a) gcc -std=c11 -Wall -pedantic-errors -o prog.exe math2.c glavni.c
b) gcc -std=c11 -Wall -pedantic-errors -c math2.c
   gcc -std=c11 -Wall -pedantic-errors -c glavni.c
   gcc -o prog.exe math2.o glavni.o
```

2. Program provjerava parnost preko tipkovnice unesenog cijelog broja. Radi se o neizravnoj rekurziji: funkcija jestParan poziva funkciju jestNeparan, ova opet funkciju jestParan i tako dalje.

   Program se ne može prevesti jer u trenutku prevođenja funkcije jestNeparan prevodiocu nije dostupna informacija o tipu i parametrima funkcije jestParan. Promjena redoslijeda funkcija ne bi pomogla. Potrebno je dodati prototipove funkcija jestNepraran i jestParan (a minimalno prototip funkcije jestParan).