**1.**

/// 1. feladat: Írjon és hívjon meg egy fgv1() nevû függvényt, bemenõ érték paraméter átadásával,

/// melyben számítsa ki két párhuzamosan kapcsolt Ohmos ellenállás eredõ értékét!

/// Az ellenállások értékei pozitív egészek: R1=10 Ohm és R2=20 Ohm , az eredmény valós érték.

/// A függvény bemenõ paraméterei az ellenállások értékei legyenek.

/// A feladatot az értékek hibakezelésével kell megoldani, hiba esetén a visszatérési érték -100!

/// Az eredményt, azaz a fgv1() visszatérési értékét teljes decimális formátumban is,

/// és 3 tizedesre kerekítve is írassa ki a main() fõfüggvényben, új sorba a képernyõre!

#include <stdio.h>

#include <stdlib.h>

double fgv1(int r1, int r2);//fgv.deklaracio

int main()

{

double y1 = 0;

y1 = fgv1( 10 , 20 );

printf("1.a. feladat: R eredo [Ohm] = %lf \n", y1);

printf("1.b. feladat: R eredo [Ohm] = %.3lf \n\n", y1);//fgv.hivas

return 0;

}

double fgv1(int r1, int r2)//fgv.definicio

{

double e=0;

if( ((r1+r2)==0) || r1<0 || r2<0 )

{

e=-100;

}

else

{

e=(r1\*r2)/ (double) (r1+r2);

}

return e;

}

**2**.

/// 2. feladat: Írjon és hívjon meg egy fgv2() nevû függvényt, bemenõ érték paraméter átadásával,

/// melyben számítsa ki egy ellenálláson mért egyenfeszültség és egyenáram értékébõl Ohm-törvény

/// felhasználásával az ellenállás értékét!

/// A feszültség és áramérték valósak lehetnek: U=15,2Volt, I=3,2Amper , az eredmény valós érték.

/// A függvény bemenõ paraméterei a feszültség és áramérték legyen.

/// A feladatot az értékek hibakezelésével kell megoldani, hiba esetén a visszatérési érték -200!

/// Az eredményt, azaz a fgv2() visszatérési értékét teljes decimális formátumban is,

/// és 3 tizedesre kerekítve is írassa ki a main() fõfüggvényben, új sorba a képernyõre!

#include <stdio.h>

#include <stdlib.h>

double fgv2(double u, double i);//fgv.deklaracio

int main()

{

double y2 = 0;

y2 = fgv2( 15.2 , 3.2 );//fgv.hivas

printf("2.a. feladat: R erteke [Ohm] = %lf \n", y2);

printf("2.b. feladat: R erteke [Ohm] = %.3lf \n\n", y2);

return 0;

}

double fgv2(double u, double i)

{

double e=0;

if( i==0 )

{

e=-200;

}

else

{

e=u/i;

if(e<0) e=e\*(-1);

}

return e;

}

**3.**

/// 3. feladat: Írjon és hívjon meg egy fgv3() nevû függvényt, bemenõ érték paraméter átadásával,

/// melyben egy izzólámpa áramfelvételét határozza meg!

/// Az izzólámpát most Ohmos terhelésnek tekintjük.

/// Az ismert adatok pozitív egészek; feszültség: Ueff = 230 Volt, teljesítmény: P = 100 Watt,

/// az eredmény valós érték.

/// A függvény bemenõ paraméterei a feszültség és teljesítmény értéke legyen.

/// A feladatot az értékek hibakezelésével kell megoldani, hiba esetén a visszatérési érték -300!

/// Az eredményt, azaz a fgv3() visszatérési értékét teljes decimális formátumban is,

/// és 3 tizedesre kerekítve is írassa ki a main() fõfüggvényben, új sorba a képernyõre!

/// Képletek; P=U\*I , R=U/I

**#include <stdio.h>**

**#include <stdlib.h>**

**double fgv3(int u, int p);//fgv.deklaracio**

**int main()**

**{**

**double y3 = 0;**

**y3 = fgv3( 230 , 100 );**

**printf("3.a. feladat: Ieff (aramfelvetel) [A] = %lf \n", y3);**

**printf("3.b. feladat: Ieff (aramfelvetel) [A] = %.3lf \n\n", y3);**

**return 0;**

**}**

**double fgv3(int u, int p)**

**{**

**double e=0;**

**if( (u==0) || u<0 || p<0 )**

**{**

**e=-300;**

**}**

**else**

**{**

**e=p/(double)u;**

**}**

**return e;**

**}**

**4.**

/// 4. feladat: Írjon és hívjon meg egy fgv4() nevû függvényt, bemenõ érték paraméter átadásával,

/// melyben egy izzólámpa ellenállását határozza meg!

/// Az izzólámpát most Ohmos terhelésnek tekintjük.

/// Az ismert adatok pozitív egészek; feszültség: Ueff = 230 Volt, teljesítmény: P = 75 Watt,

/// az eredmény valós érték.

/// A függvény bemenõ paraméterei a feszültség és teljesítmény értéke legyen.

/// A feladatot az értékek hibakezelésével kell megoldani, hiba esetén a visszatérési érték -400!

/// Az eredményt, azaz a fgv4() visszatérési értékét teljes decimális formátumban is,

/// és 3 tizedesre kerekítve is írassa ki a main() fõfüggvényben, új sorba a képernyõre!

**#include <stdio.h>**

**#include <stdlib.h>**

**double fgv4(int u, int p);//fgv.deklaracio**

**int main()**

**{**

**double y4 = 0;**

**y4 = fgv4( 230 , 75 );**

**printf("4.a. feladat: R [Ohm] = %lf \n", y4);**

**printf("4.b. feladat: R [Ohm] = %.3lf \n\n", y4);**

**return 0;**

**}**

**double fgv4(int u, int p)**

**{**

**double e=0;**

**if( (u==0) || u<0 || p<0 )**

**{**

**e=-400;**

**}**

**else**

**{**

**e=(u\*u)/(double)p;**

**}**

**return e;**

**}**

**5.**

/// 5. feladat: Írjon és hívjon meg egy fgv5() nevû függvényt, bemenõ érték paraméter átadásával,

/// melyben kiszámolja a Ganzuniv 4 mérõmûszer relatív formában megadott véletlen hibáját (±h=?%),

/// ha a mûszerrel egyenfeszültséget mérünk.

/// A mért feszültség és a pontossági osztály valós értékek; Um = 2,9 Volt , ±hpo = 1,5% !

/// A méréshatár pozitív egész érték: Umh = 3 Volt !

/// A függvény bemenõ paraméterei a megadott értékek legyenek (Um , ±hpo, Umh).

/// A feladatot az értékek hibakezelésével kell megoldani, hiba esetén a visszatérési érték -500!

/// Az eredményt, azaz a fgv5() visszatérési értékét teljes decimális formátumban is,

/// és 3 tizedesre kerekítve is írassa ki a main() fõfüggvényben, új sorba a képernyõre!

/// Képlet: ±h = ±hpo \* Umh / Um .

**#include <stdio.h>**

**#include <stdlib.h>**

**double fgv5(double um, double hpo, int umh);//fgv.deklaracio**

**int main()**

**{**

**double y5 = 0;**

**y5 = fgv5( 2.9 , 1.5 , 3 );**

**printf("5. feladat: +-h [%c] = +- %lf \n", '%', y5);**

**printf("5. feladat: +-h [%c] = +- %.3lf \n\n", '%', y5);**

**return 0;**

**}**

**double fgv5(double um, double hpo, int umh)**

**{**

**double e=0;**

**if( um==0 || umh<=0 )**

**{**

**e=-500;**

**}**

**else**

**{**

**e = hpo \* umh / um ;**

**if(e<0) e=e\*(-1);**

**}**

**return e;**

**}**

**6.**

/// 6. feladat: Írjon és hívjon meg egy fgv6() nevû függvényt, bemenõ érték paraméter átadásával,

/// melyben összegzi [-20,32] közötti ZÁRT intervallumban azon EGÉSZ SZÁMÉRTÉKEKET,

/// melyekre igaz, hogy az egyes számértékek oszthatók 7-el, de nem oszthatók 3-mal!

/// A függvény bemenõ paraméterei az intervallum határai legyenek.

/// Az eredmény egész, integer (int) típusú legyen!

/// Az eredményt, azaz a fgv6() visszatérési értékét hexadecimális formátumban írassa ki

/// a main() fõfüggvényben, új sorba a képernyõre!

/// Ezt a hexadecimális eredményt prefixum (0x) nélkül írja be a MOODLE ablakba!

**//-14,-7,7,14,28=28=1C**

**#include <stdio.h>**

**#include <stdlib.h>**

**int fgv6(int ksz, int vsz);//fgv.deklaracio**

**int main()**

**{**

**int y6 = 0;**

**y6 = fgv6( -20 , 32);**

**printf("6. feladat: Megfelelo szamok osszege (hex.): sum = 0x %x \n\n", y6);**

**return 0;**

**}**

**int fgv6(int ksz, int vsz)**

**{**

**int sum=0;**

**for(ksz; ksz<=vsz; ksz++)**

**{**

**if( ((ksz%7)==0) && ((ksz%3)!=0) )**

**{**

**sum=sum+ksz;**

**}**

**}**

**return sum;**

**}**

**7.**

/// 7. feladat: Írjon és hívjon meg egy fgv7() nevû függvényt, bemenõ érték paraméter átadásával,

/// melyben a [10,30] közötti EGÉSZ SZÁMÉRTÉKÛ ZÁRT intervallumon az alábbi mûveletsort hajtja

/// végre;

/// Az intervallumban található 3-al osztható egész számok összegét megszorozza

/// az 5-tel osztható egész számok összegével (WHILE-ciklus felhasználásával)!

/// A függvény bemenõ paraméterei az intervallum határai legyenek.

/// Az eredményt, azaz a fgv7() visszatérési értékét decimális formátumban írassa ki

/// a main() fõfüggvényben, új sorba a képernyõre!

/// Ezt a decimális eredményt írja be a MOODLE ablakba!

#include <stdio.h>

#include <stdlib.h>

int sf(int ksz,int vsz);//fgv.deklaracio

int main()

{

int kez=10,veg=30,sz;

sz=sf(kez,veg);//fgv.hivas

printf("A fgv.visszateresi erteke:%d",sz);

return 0;

}

int sf(int ksz,int vsz)//fgv.definicio

{

int osszeg3=0, osszeg5=0, szorzat=1;

while (ksz<=vsz)

{

if(ksz%3==0)

{

osszeg3=osszeg3+ksz;

}

if(ksz%5==0)

{

osszeg5=osszeg5+ksz;

}

ksz++;//ksz=ksz+1

}

szorzat=osszeg3\*osszeg5;

return szorzat;

}

**8.**

/// 8. feladat: Írjon és hívjon meg egy fgv8() nevû függvényt, bemenõ érték paraméter átadásával,

/// melyben megszámolja [2,20] közötti ZÁRT intervallumban azon EGÉSZ SZÁMÉRTÉKEKET,

/// melyekre igaz, hogy 0. biten 1-es található, 2. biten 0-ás található.

/// A függvény bemenõ paraméterei az intervallum határai legyenek.

/// Az eredményt (darabszámot), azaz a fgv7() visszatérési értékét decimális formátumban írassa ki

/// a main() fõfüggvényben, új sorba a képernyõre!

/// Ezt a decimális eredményt írja be a MOODLE ablakba!

**#include <stdio.h>**

**#include <stdlib.h>**

**int fgv8(int ik, int iv);//fgv.deklaracio**

**int main()**

**{**

**int y8 = 0;**

**y8 = fgv8( 2 , 20 );**

**printf("8. feladat: Megfelelo szamok darabszama (dec.): db = %d \n\n", y8);**

**return 0;**

**}**

**int fgv8(int ik, int iv)**

**{**

**int i=0, db=0;**

**for(i=ik;i<=iv;i++)**

**{**

**if( (i & 0b0101) == 0b0001 )**

**{**

**db++;**

**}**

**}**

**return db;**

**}**

**#include <stdio.h>**

**#include <stdlib.h>**

**9.**

/// 9. feladat: Írjon és hívjon meg egy fgv9() nevû függvényt, bemenõ érték paraméter átadásával,

/// melyben megszámolja, hogy hány darab EGÉSZ ÉRTÉKÛ négyzetszám van a [10,40] közötti ZÁRT

/// intervallumban.

/// A függvény bemenõ paraméterei az intervallum határai legyenek.

/// Az eredményt (darabszámot), azaz a fgv9() visszatérési értékét decimális formátumban írassa ki

/// a main() fõfüggvényben, új sorba a képernyõre!

/// Ezt a decimális eredményt írja be a MOODLE ablakba!

**int fgv9(int ik, int iv);//fgv.deklaracio**

**int main()**

**{**

**int y9 = 0;**

**y9 = fgv9( 10 , 40 );**

**printf("9. feladat: A negyzetszamok (dec.): db = %d \n\n", y9);**

**return 0;**

**}**

**int fgv9(int ik, int iv)**

**{**

**int i=0, db=0;**

**while( 1 )**

**{**

**if( (i\*i)>=ik && (i\*i)<=iv)**

**{**

**db++;**

**}**

**i++;**

**if ( (i\*i)>iv) break;**

**}**

**return db;**

**}**