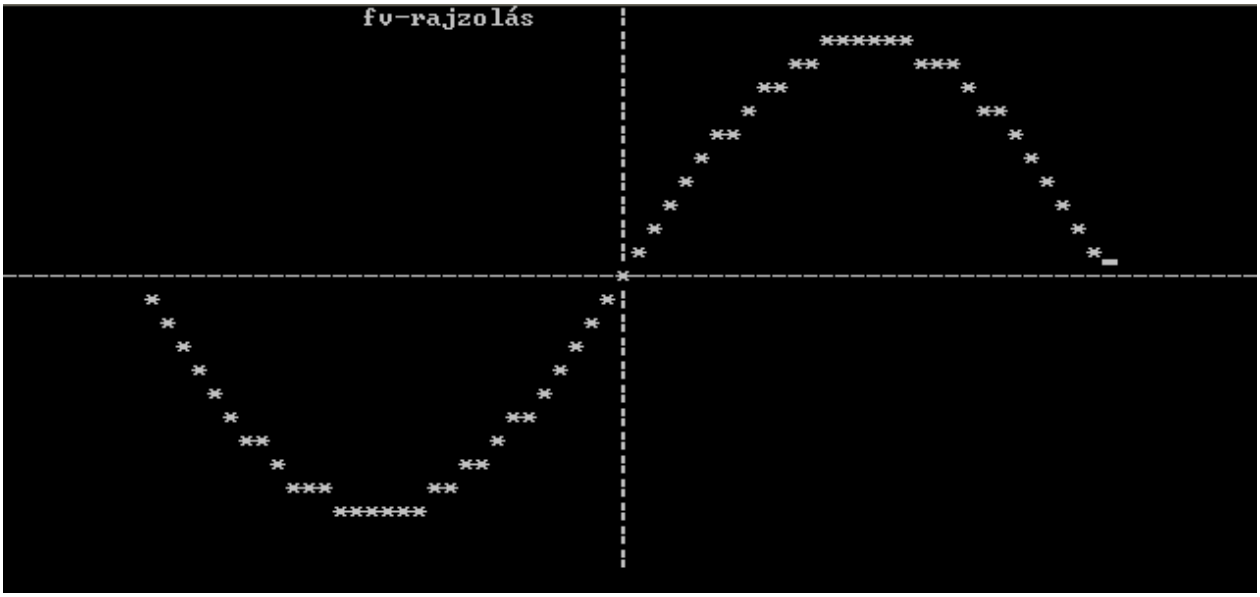
Az első program

Figyeljük meg, miként készül el az informális feladatszövegből először a formális leírása, majd a megoldó program algoritmusa és (Pascal) kódja! Vegyük észre, hogy az egyes lépésekben felhasználjuk az előző lépés(ek)ben már rögzített döntéseinket!

Feladat:

Az f(x) függvény kirazolása karakteres képernyön (bambán).
Az f(x):=a*sin(b*x+c)+d; x∈[x1..x2].

A feladat értelmezéséhez próbáljuk ki a kódot. (ElsoPrg0.exe)



Specifikáció:

Be: A,B,C,D,X1,X2∈ℝ *[fv-paraméterek]*
Ki: Kép∈$\mathbb{C}^{MaxS*MaxO}$, MaxO=25, MaxS=80 *[Kép=képernyő, mint karakter-mátrix]*
Ef: X1<X2 ∧
[KK origója a bal-felső sarokban ∧ $KK_{egység}=VK_{egység}$][1]
[KK=képernyő-koordinátarendszer, VK=Világ-koordinátarendszer]
Uf: S0=MaxS Div 2 ∧ O0=MaxO Div 2 *[a VK képe KK-ban, az origója a képernyő közepén]* ∧
(∀s∈[1..MaxS]\{S0}: f(0) ≠s) ⇒ Kép(O0,s)='|' *[a képernyő O0 oszlopában '|', ha éppen a fv nem takarja el]* ∧
(∀o∈[1..MaxO]\{O0}: f(o–O0) ≠0) ⇒ Kép(o,S0)='|' *[a képernyő S0 sorában '–', ha éppen a fv nem takarja el]* ∧
(0∈[X1..X2] ∧ f(0)≠0) ⇒ Kép(O0,S0)='+' *[a VK origójának képében '+', ha éppen a fv nem takarja el]* ∧
(∀x∈[X1..X2]: (x+O0,S0–f(x))∈[1..MaxO]×[1..MaxS]) ⇒ Kép(x+O0,S0-f(x))='*' *[a megfelelö helyen jelenik meg a fv érték]*
Def: f:[x1..x2]→ℝ *[fv-szignatúra]*
f(x)=A*sin(B*x+C)+D

[1] A zárójelezett feltételek nem a bemenettel szemben fogalmaz meg elvárandókat, hanem a megjelenítést végző képernyő szervezésről árul el figyelembe veendőket.

Algoritmus:

Az algoritmust felülről lefelé tervezéssel végezzük! Kezdjük a legfelsőbb szintű, ún. „főprogrammal” (és az ún. globális adatokkal), majd finomítsuk a felhasznált, de még nem részletezett tevékenységeket (eljárásokat, függvényeket)! Érdemes összeszedegetni, hogy az algoritmus mely részén a specifikáció mely kijelentését használunk fel közvetlenül/közvetve.

```
Program ElsőProgram:
  [a globális adatok:]
  Konstans
    MaxS=24; MaxO=80
    CsillagJel='*'; VizszintJel='-'; FüggőJel='|'; PluszJel='+'
    TeliJel='▒'; [177-es kódú karakter][2]
  Változó
    A,B,C,D,X1,X2:Valós
   [Kép:Tömb(1..MaxS,1..MaxO:Karakter)]
    O0,S0:Egész

  [a főprogram:]
  S0:=MaxS Div 2; O0:=MaxO Div 2
  Beolvasas
  Kirajzolas
Program vége.

[a specifikáció részeként:]
Függvény f(Konstans x:Valós):Valós
  f:=A*sin(B*x+C)+D
Függvény vége.

[a főprogram finomításaként:]
Eljárás Beolvasás:
  Be: A,B,C,D
  Be: X1,X2 [X1<X2]
Eljárás vége.

Eljárás Kirajzolás:
  Változó
    x[ciklusváltozó]:Egész

  X_Tengely(S0); Y_Tengely(O0); Origó(O0,S0)
  Ciklus x=Kerekít(X1)-től Kerekít(X2)-ig
    FvErtekKirajzolas(x,Kerekít(f(x)))
  Ciklus vége
Eljárás vége.

[a Kirajzolás finomításaként:]
Eljárás X_Tengely(Konstans O0:Egész):
  Változó
    x:Egész

  Ki(1,S0): [pozícionálás]
  Ciklus x=1-től MaxO-ig
    Ki: VizszintJel
  Ciklus vége
Eljárás vége.
```

[2] Egy későbbi módosítás miatt kerül ide. Egyelőre figyelmen kívül hagyható.

```
Eljárás Y_Tengely(Konstans S0:Egész):
  Változó
    y:Egész
  Ciklus y=1-től MaxS-ig
    Ki(O0,y): FüggőJel[3]
  Ciklus vége
Eljárás vége.

Eljárás Origó(Konstans O0,S0:Egész):
  Ki(O0,S0): PluszJel
Eljárás vége.

Eljárás FvErtekKirajzolas(Konstans x,y:Egész):
  Változó
    o,s:Egész
  o:=O0+x; s:=S0-y
  Ha (o<=MaxO) és (o>0) és (s<=MaxS) és (s>0) akkor
    [a pont rajta van a képernyön]
    Ki(o,s): CsillagJel
  Elágazás vége
Eljárás vége.
```

Kód:

A kódolás során felvetődő, algoritmusban nem szereplő „újdonságokat” pirossal jelöltük. Bár a kódon nem látszik, de tudjunk róla, hogy célszerű a kódolást is felülről lefelé sorrendben haladva végezni.

```
Program ElsoProgram;
(*
  Feladat: az f(x) fv kirazolása karakteres képernyön (bambán).
           Az f(x):=y*sin(b*x+c)+d; x ELEME [x1..x2].
  Be: A,B,C,D,X1,X2 ELEME Valós
  Ki: Kép ELEME Karakter^MaxS*MaxO, MaxO=25,MaxS=80,
  Ef: X1<X2
      ÉS a képernyö koordinátarendszer (KK) origója a bal-felsö sarokban
      ÉS a KK egysége=világ koordinátarendszer (VK) egysége
  Uf: az origó a képernyö közepén:
        O0=MaxO Div 2 ÉS S0=MaxS Div 2 ÉS
      a megfelelö helyen jelenik meg a fv érték:
        Kép(S,O)='*' <=> S=S0-f(x) ÉS O=O0+x ÉS
      a képernyő S0 sorában '-':
        MINDEN x ELEME [1..MaxS]\{O0}: Kép[S0,x]='-' ÉS
      a képernyő O0 oszlopában '|':
        MINDEN y ELEME [1..MaxO]\{S0}: Kép[y,O0]='|' ÉS
      a VK origójának képében '+':
        Kép[S0,O0]='+'
*)
  Uses
    {Newdelay[4],}Crt;
```

[3] A „Ki(s,o):valami” művelet a képernyő s. sorának o. karakterpozíciójától kezdődően írja a valamit. A művelet után a kurzor a valami utánra áll.

```
Const
  cim='fv-rajzolás';
  MaxS=24; MaxO=80;
  CsillagJel='*'; VizszintJel='-'; FuggoJel='|'; PluszJel='+';
  TeliJel='▒'; {177-es kódú karakter}
Var
  A,B,C,D,X1,X2:Real;
 {Kép:Array [1..MaxS,1..MaxO] of Char;}
  O0,S0:Integer;

Procedure UjLap(Const cim:String);
Begin
  ClrScr; Writeln(cim:(MaxO-Length(cim)) Div 2);
End;

Function f(Const x:Real):Real;
Begin
  f:=A*sin(B*x+C)+D
End;

Procedure Beolvasas;
Begin
  Write('A fv. paraméterei (A,B,C,D; pl. 10 .1 0 0):');
  Readln(A,B,C,D); {pl. 10 .1 0 0}
  Repeat
    Writeln('A fv. értelmezési tartománya (X1,X2; pl. -30 30):');
    Readln(X1,X2); {pl. -30 30}
  Until X1<X2;
End;

Procedure X_Tengely(Const S0:Integer);
  Var
    x{ciklusváltozó}:Integer;
Begin
  GotoXY(1,S0);
  For x:=1 to MaxO do Write(VizszintJel);
End;

Procedure Y_Tengely(Const O0:Integer);
  Var
    y{ciklusváltozó}:Integer;
Begin
  For y:=1 to MaxS do
  Begin
    GotoXY(O0,y); Write(FuggoJel);
  End;
End;

Procedure Origo(Const O0,S0:Integer);
Begin
  GotoXY(O0,S0); Write(PluszJel);
End;
```

[4] A Turbo Pascal esetében.

```
  Procedure FvErtekKirajzolas(Const x,y:Integer);
    Var
      o,s:Integer;
  Begin
    o:=O0+x; s:=S0-y;
    If (o<=MaxO) and (o>0) and (s<=MaxS) and (s>0) then
    Begin {a pont rajta van a képernyön}
      GotoXY(o,s); Write(CsillagJel)
    End;
  End;

  Procedure Kirajzolas;
    Var
      x{ciklusváltozó}:Integer;
  Begin
    X_Tengely(S0); Y_Tengely(O0);
    For x:=Round(X1) to Round(X2) do
    Begin
      FvErtekKirajzolas(x,Round(f(x)));
    End;
  End;

Begin
  UjLap(cim);
  O0:=MaxO Div 2; S0:=MaxS Div 2;
  Beolvasas;
  UjLap(cim);
  Kirajzolas;
  ReadKey;
End.
```

A függvény egyes értékeinek kirajzolását kicsit szebbé lehet tenni azzal, ha nem egyetlen ponttal, hanem egy, a pontig tartó függőleges pontsorral jelöljük. Ehhez a pontrajzolás helyett egy ciklusos pontrajzolás kell.

```
  Eljárás FvErtekKirajzolas(Konstans x,y:Egész);
    Változó
      o,s,ss,tol,ig:Egész

    o:=O0+x; s:=S0-y
    Ha o<=MaxO és o>0 akkor [az oszlop a képernyőn van]
      Ha  s<S0  akkor tol:=s; ig:=S0
            különben tol:=S0; ig:=s
      Ciklus ss=tol-től ig-ig [felülről lefelé rajzolás]
        Ha ss<=MaxS és ss>0 akkor [a pont rajta van a képernyön]
          Ki(o,ss): TeliJel
        Elágazás vége
      Ciklus vége
    Elágazás vége
  Eljárás vége.
```

Kódja:

```
  Procedure FvErtekKirajzolas(Const x,y:Integer);
    Var
      o,s,ss,tol,ig:Integer;
```

```
Begin
  o:=O0+x; s:=S0-y;
  If (o<=MaxO) and (o>0) then {az oszlopot lehet rajzolni}
  Begin
    If s<S0 then
    Begin
      tol:=s; ig:=S0
    End
      Else
    Begin
      tol:=S0; ig:=s
    End;
    For ss:=tol to ig do
    Begin
      If (ss<=MaxS) and (ss>0) then {a pont rajta van a képernyön}
      Begin
        GotoXY(o,ss); Write(TeliJel)
      End;
    End;
  End;
End;
```

Próbáljuk ki a módosított kódot is! (ElsoPrg1.exe)