

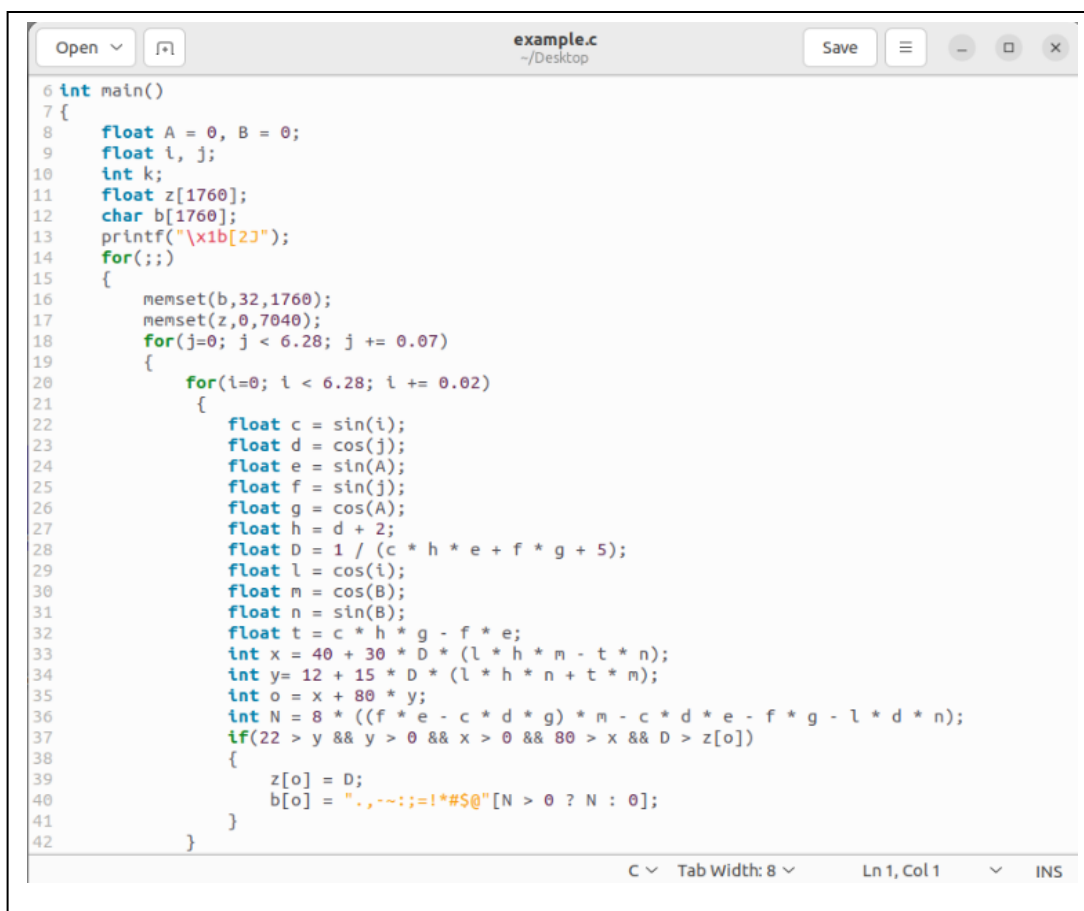
سوال 1

ابتدا برای کامپایلر کد C، نیاز به یک کامپایلر C داریم. از آنجایی که GCC به صورت دیفالت در Ubuntu ای که دانلود کردیم نصب شده، از این کامپایلر استفاده می کنیم.

```
hidden@hidden-virtual-machine:~/Desktop$ sudo apt install gcc
[sudo] password for hidden:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc is already the newest version (4:12.2.0-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 162 not upgraded.
hidden@hidden-virtual-machine:~/Desktop$
```

می بینیم که از قبل نصب بود و نیازی به نصب دوباره آن نیست.

حال، با استفاده از دستور touch، یک فایل به نام example.c درست می کنیم و با gedit آن را باز کرده و کد را در آن می نویسیم:



```
example.c
~/Desktop

6 int main()
7 {
8     float A = 0, B = 0;
9     float i, j;
10    int k;
11    float z[1760];
12    char b[1760];
13    printf("\x1b[2J");
14    for(;;)
15    {
16        memset(b,32,1760);
17        memset(z,0,7040);
18        for(j=0; j < 6.28; j += 0.07)
19        {
20            for(i=0; i < 6.28; i += 0.02)
21            {
22                float c = sin(i);
23                float d = cos(j);
24                float e = sin(A);
25                float f = sin(j);
26                float g = cos(A);
27                float h = d + 2;
28                float D = 1 / (c * h * e + f * g + 5);
29                float l = cos(i);
30                float m = cos(B);
31                float n = sin(B);
32                float t = c * h * g - f * e;
33                int x = 40 + 30 * D * (l * h * m - t * n);
34                int y = 12 + 15 * D * (l * h * n + t * m);
35                int o = x + 80 * y;
36                int N = 8 * ((f * e - c * d * g) * m - c * d * e - f * g - l * d * n);
37                if(22 > y && y > 0 && x > 0 && 80 > x && D > z[o])
38                {
39                    z[o] = D;
40                    b[o] = ".,-~:;=!*#$@"[N > 0 ? N : 0];
41                }
42            }
43        }
44    }
```

```

hidden@hidden-virtual-machine: ~/Desktop$ touch example.c
hidden@hidden-virtual-machine: ~/Desktop$ gedit example.c
hidden@hidden-virtual-machine: ~/Desktop$ gcc example.c -o example.o
/usr/bin/ld: /tmp/ccKfFD0.o: in function `main':
example.c:(.text+0xcd): undefined reference to `sin'
/usr/bin/ld: example.c:(.text+0xf4): undefined reference to `cos'
/usr/bin/ld: example.c:(.text+0x11b): undefined reference to `sin'
/usr/bin/ld: example.c:(.text+0x142): undefined reference to `sin'
/usr/bin/ld: example.c:(.text+0x169): undefined reference to `cos'
/usr/bin/ld: example.c:(.text+0x1fb): undefined reference to `cos'
/usr/bin/ld: example.c:(.text+0x222): undefined reference to `cos'
/usr/bin/ld: example.c:(.text+0x249): undefined reference to `sin'
collect2: error: ld returned 1 exit status
hidden@hidden-virtual-machine: ~/Desktop$ gedit example.c
^C
hidden@hidden-virtual-machine: ~/Desktop$ gcc example.c -o example.o -lm
hidden@hidden-virtual-machine: ~/Desktop$

```

همانطور که در عکس بالا مشخص است، در ابتدا linker قابلیت پیدا کردن فانکشن های ریاضی sin و cos را که در هدر math.h تعریف شده اند را نداشت که با اضافه کردن فیلگ -lm، کتابخانه math را به برنامه خود link کردیم. این کتابخانه به دلایل تاریخی، به صورت اتوماتیک لینک نمی شود. حال با اجرای دستور ./example.o، برنامه خود را اجرا می کنیم:

خروجی برنامه، دوناتی است که به دور خود می چرخد.

منبع: <https://www.a1k0n.net/2011/07/20/donut-math.html>