```
  1: //================================================================================================
  2: //================================================================================================
  3: // PROGRAMA                        : pe27.cpp
  4: // FUNCION                         : Conversion de coordenadas polares a rectangulares.
  5: //                                   processor: x86
  6: // REALIZADO POR                   : Prof. Juan Ju?rez Fuentes
  7: // COMPILADOR EN EL QUE SE EJECUTO: MDev-C++ version 5.5.2
  8: // FECHA                           : 20240605114143
  9: //================================================================================================
 10: // LIBRERIAS
 11: //================================================================================================
 12: #include <stdio.h>
 13: #include <math.h>
 14: //================================================================================================
 15: // PROTOTIPO DE FUNCIONES
 16: //================================================================================================
 17: void polar_a_rectangular(double r, double theta, double *x, double *y);
 18: //================================================================================================
 19: // PRINCIPAL
 20: //================================================================================================
 21: int main()
 22: {
 23:     double r ;
 24:     double theta;
 25:     double x;
 26:     double y;
 27:
 28:     printf("\n");
 29:     printf("================================================================================\n");
 30:     printf(" \t\tPROGRAMA HIBRIDO C-ENSAMBLADOR \n");
 31:     printf(" \t\tQUE CONVIERTE COORDENADAS POLARES A RECTANGULARES\n");
 32:     printf("================================================================================\n\n");
 33:
 34:     // Dar datos.
 35:     printf("\tDar el valor de r: ");
 36:     scanf("%lf",&r);
 37:     printf("\tDar el valor de theta (en radianes): ");
 38:     scanf("%lf",&theta);
 39:
 40:     // Convertir a coordenadas rectangulares
 41:     polar_a_rectangular(r, theta, &x, &y);
 42:
 43:     // Imprimir el resultado
```

```
44:     printf("\n\tCoordenadas rectangulares: \n\tx = %lf \n\ty = %lf\n", x, y);
45:
46:     return 0;
47: }
48: //============================================================================================
49: //  FUNCIONES
50: //============================================================================================
51: // FunciÃ³n para convertir coordenadas polares a rectangulares
52: void polar_a_rectangular(double r, double theta, double *x, double *y)
53: {
54:     double cos_theta, sin_theta;
55:
56:     // Usar ensamblador inline para calcular cos(theta) y sin(theta)
57:     asm (
58:         "fsincos"        // Calcula cos(theta) y sin(theta)
59:         : "=t"(cos_theta), "=u"(sin_theta)
60:         : "0"(theta)
61:     );
62:
63:     // Usar ensamblador inline para calcular x = r * cos(theta) y y = r * sin(theta)
64:     asm (
65:         "movq %2, %%xmm0;"       // Mueve r a xmm0
66:         "movq %3, %%xmm1;"       // Mueve cos_theta a xmm1
67:         "mulsd %%xmm1, %%xmm0;"  // xmm0 = xmm0 * xmm1 (x = r * cos_theta)
68:         "movq %%xmm0, %0;"       // Mueve el resultado a x
69:         : "=m"(*x)
70:         : "m"(*x), "m"(r), "m"(cos_theta)
71:         : "%xmm0", "%xmm1"
72:     );
73:
74:     asm (
75:         "movq %2, %%xmm0;"       // Mueve r a xmm0
76:         "movq %3, %%xmm1;"       // Mueve sin_theta a xmm1
77:         "mulsd %%xmm1, %%xmm0;"  // xmm0 = xmm0 * xmm1 (y = r * sin_theta)
78:         "movq %%xmm0, %0;"       // Mueve el resultado a y
79:         : "=m"(*y)
80:         : "m"(*y), "m"(r), "m"(sin_theta)
81:         : "%xmm0", "%xmm1"
82:     );
83: }
84: //============================================================================================
85: //============================================================================================
86: //============================================================================================
```

```
87: //============================================================================================
```