**Nama = Ancas Nuzul Nur Hidayah**

**NIM = L200150068**

**Kelas = B**

**TUGAS 2 ALGORITMA DAN STRUKTUR DATA**

1. **MENAMPILKAN DATA ASCII**
2. **Kode Program**

print("Daftar Kode ASCII")

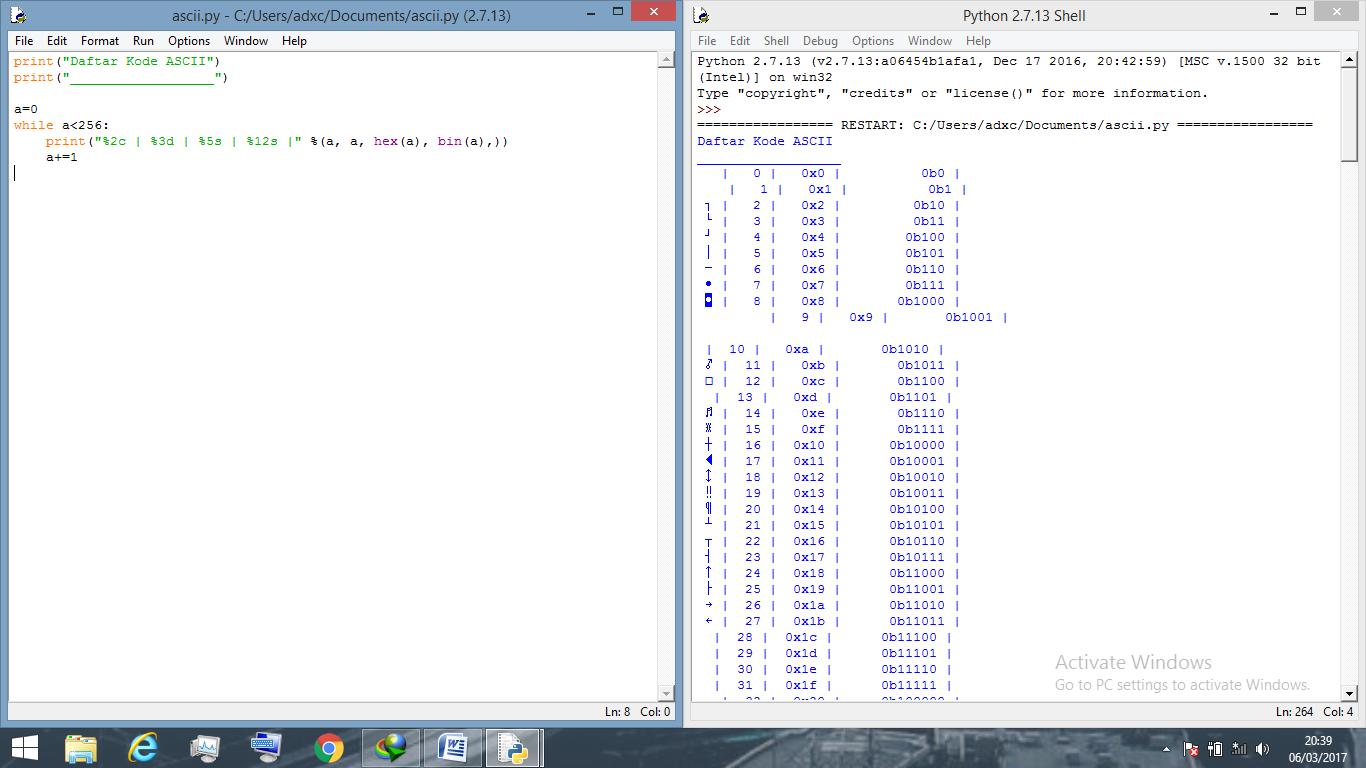
print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

a=0

while a<256:

print("%2c | %3d | %5s | %12s |" %(a, a, hex(a), bin(a),))

a+=1

1. **Screenshot**
2. **FUNGSI DALAM MODUL MATH**

* acos(...)

Return the arc cosine (measured in radians) of x.

* acosh(...)

Return the hyperbolic arc cosine (measured in radians) of x.

* asin(...)

Return the arc sine (measured in radians) of x.

* asinh(...)

Return the hyperbolic arc sine (measured in radians) of x.

* atan(...)

Return the arc tangent (measured in radians) of x.

* atan2(y, x)

Return the arc tangent (measured in radians) of y/x.

Unlike atan(y/x), the signs of both x and y are considered.

* atanh(x)

Return the hyperbolic arc tangent (measured in radians) of x.

* ceil(x)

Return the ceiling of x as a float. This is the smallest integral value >= x.

* copysign(x, y)

Return x with the sign of y.

* cos(x)

Return the cosine of x (measured in radians).

* cosh(x)

Return the hyperbolic cosine of x.

* degrees(x)

Convert angle x from radians to degrees.

* erf(x)

Error function at x.

* erfc(x)

Complementary error function at x.

* exp(x)

Return e raised to the power of x.

* expm1(x)
* Return exp(x)-1.

This function avoids the loss of precision involved in the direct evaluation of exp(x)-1 for small x.

* fabs(x)

Return the absolute value of the float x.

* factorial(x) -> Integral

Find x!. Raise a ValueError if x is negative or non-integral.

* floor(x)

Return the floor of x as a float. This is the largest integral value <= x.

* fmod(x, y)

Return fmod(x, y), according to platform C. x % y may differ.

* frexp(x)

Return the mantissa and exponent of x, as pair (m, e). m is a float and e is an int, such that x = m \* 2.\*\*e. If x is 0, m and e are both 0. Else 0.5 <= abs(m) < 1.0.

* fsum(iterable)

Return an accurate floating point sum of values in the iterable. Assumes IEEE-754 floating point arithmetic.

* gamma(x)

Gamma function at x.

* hypot(x, y)

Return the Euclidean distance, sqrt(x\*x + y\*y).

* isinf(x) -> bool

Check if float x is infinite (positive or negative).

* isnan(x) -> bool

Check if float x is not a number (NaN).

* ldexp(x, i)

Return x \* (2\*\*i).

* lgamma(x)

Natural logarithm of absolute value of Gamma function at x.

* log(x[, base])

Return the logarithm of x to the given base. If the base not specified, returns the natural logarithm (base e) of x.

* log10(x)

Return the base 10 logarithm of x.

* log1p(x)

Return the natural logarithm of 1+x (base e). The result is computed in a way which is accurate for x near zero.

* modf(x)

Return the fractional and integer parts of x. Both results carry the sign of x and are floats.

* pow(x, y)

Return x\*\*y (x to the power of y).

* radians(x)

Convert angle x from degrees to radians.

* sin(x)

Return the sine of x (measured in radians).

* sinh(x)

Return the hyperbolic sine of x.

* sqrt(x)

Return the square root of x.

* tan(x)

Return the tangent of x (measured in radians).

* tanh(x)

Return the hyperbolic tangent of x.

* trunc(x:Real) -> Integral

Truncates x to the nearest Integral toward 0. Uses the \_\_trunc\_\_ magic method.