Programming Translation Table

Operation	English Steps	Algorithm Example	C Statement Example
Comment (not an operation)	N/A	{This is a comment}	/*This is a comment*/
Declare an Integer variable	Declare variable 'a' of type integer	var a:Integer;	int a;

Declare a fraction variable	Declare variable 'a' of type real	var a:Float;	float a;
			//or
			double a;

Declare a character variable	Declare variable 'a' of type character	var a:Character;	char a;
Assignment	Assign value 5 to variable 'a'	a := 5;	a=5;
Addition and assignment	Assign value of adding variable 'a' and variable 'b' to variable 'c'	c := a+b;	c=a+b;
Subtraction and assignment	Assign value of subtracting variable 'b' from variable 'a' to variable 'c'	c := a-b;	c=a-b;
Multiplication and assignment	Assign the product of variable 'a' and variable 'b' to variable 'c'	c := a*b;	c=a*b;
Division and assignment	Assign value of dividing variable 'a' by variable 'b' to variable 'c'	c := a/b;	c=a/b;
Finding remainder and assignment	Assign value of remainder obtained by dividing variable 'a' by variable 'b' to variable 'c'	c := a mod b;	c=a%b;

Find size of variable/data type	Assign size of variable 'a' to integer variable 'b'
	Assign size of integer data type to integer variable 'b'
		b := sizeof(a);
		b := sizeof(Integer);
			b = sizeof(a);
			b = sizeof(int);

Comparison* * Used in conditional or looping	Variable 'a' greater than variable 'b' Variable 'a' less than variable 'b' Variable 'a' equal to	a > b a < b a = b	a > b a < b a == b

Operation	English Steps	Algorithm Example	C Statement Example
statements only. Not used independently	variable 'b' Variable 'a' greater than or equal to variable 'b' Variable 'a' less than or equal to variable 'b' Variable 'a' not equal to variable 'b'	a >= b a <= b a <> b	a >= b a <= b a != b

Assertion of input/output	N/A: Part of problem definition	{assert a>0}	#include <assert.h>
			assert (a>0);

Read an integer	Read integer 'a' from user	read a;	#include <stdio.h>
			scanf(“%d”,&a);

Read a fraction value	Read real variable 'a' from user	read a;	#include <stdio.h>
			scanf(“%f”,&a);
			//above for float
			scanf(“%lf”,&a);
			//above for double

Read a character value	Read character variable 'a' from user	read a;	#include <stdio.h>
			scanf(“%c”,&a);

Print an integer	Print integer 'a'	write a;	#include <stdio.h>
			printf(“%d”,a);

Print a fraction value	Print real variable 'a'	write a;	#include <stdio.h>
			printf(“%f”,a);
			//above for float
			printf(“%lf”,&a);
			//above for double

Print a character value	Print character variable 'a'	write a;	#include <stdio.h>
			printf(“%c”,a);

Conditional work	if a is greater than or equal to b and a greater than or equal to c then
	print “a is greatest”
	
	else if b greater than or equal to a and and b greater than or equal to c then
	print “b is greatest”
	
	else
	print “c is greatest”
		if (a>b) and (a>c) then
		begin
		write “a is greatest”;
		end
		else (b>a) and (b>c) then
		begin
		write “b is greatest”;
		end
		else
		begin
		write “c is greatest”;
		end
			if (a>b) && (a>c)
			{
			printf(“a is greatest”);
			}
			else if (b>c) && (b>c)
			{
			printf(“b is greatest”);
			}
			else
			{
			printf(“c is greatest”);
			}

Repeat 'n' times	Repeat using index from 0 to 'n'-1,	for index := 0 to n-1, step 1 do begin	for (index = 0;index <= n-1;index++)

Programming Translation Table

Operation	English Steps	Algorithm Example	C Statement Example

	increasing value of index by 1
	print 'n'
		write n;
		end
			{
			printf("%d", n);
			}

Repeat while a condition is true	Repeat while n is greater than or equal to 0
	assign value of subtracting 1 from variable 'n' to variable 'n'
		while n >= 0 do
		begin
		n := n-1;
		end
			while (n>=0)
			{
			n = n-1;
			}

Example algorithm:

```
Algorithm addMatricies (a:MxNintegerArray, b:PXQIntegerArray, c:MxNIntegerArray, M:Integer,
N:Integer, P:Integer, Q:Integer)
var i,j:Integer;
Begin
    {assert M=P and N=Q}
    for i := 0 to M-1, step 1 do
    begin
        for j := 0 to Q-1, step 1 do
        begin
            c[i][j] := a[i][j] + b[i][j];
        end
    end
End
```

Sample C program:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>

void   addMatricies (int a[][], int b[][], int c[][], int m, int n, int p, int q)
{
        int i,j,k;
        assert((m==p)&&(n==q));
        for (i = 0; i<=m-1,i++)
        {
                for (j = 0; j<=q-1,j++)
                {
                        c[i][j] := a[i][j] * b[i][j];
                }
        }

}
/*This is where the program starts and it calls your code*/
int main ( int argc, char **argv)
{
        int a[][] = {{1,2,3},{4,5,6}};
        int b[][] = {{1,2,1},{3,4,5}};
        int c[3][2];
        addMatrices(a,b,c,3,2,3,2);
}
```