**Task 3: Solve programming problems implementing necessary loops**

1. **Calculate simple interest for 3 sets of values of principal, number of years and rate of interest.**

/\* Calculation of simple interest for 3 sets of p, n and r \*/

#include<stdio.h>

int main()

{

int p, n, count;

float r, si;

count = 1;

while (count <= 3)

{

printf ("\nEnter values of p, n and r");

scanf ("%d %d %f", &p, &n, &r);

si = p \* n \* r / 100;

printf ("Simple interest = Rs. %f", si);

count = count + 1;

}

return 0;

}

1. **Write a program to find the factorial value of any number entered through the keyboard.**

Program

/\* Calculation of factorial value of a number \*/

**include**

int main()

{

int num, i, fact;

printf (“Enter a number: ”);

scanf (“%d”, &num);

fact = i = 1;

while (i <= num)

{

fact = fact \* i;

i++;

}

printf (“Factorial value of %d = %d\n”, num, fact);

return 0;

}

1. **Write a program to find the value of one number raised to the power of another.**

**Program**

/\* Compute value of one number raised to another \*/

**include**

int main()

{

float x, power;

int y, i;

printf (“\nEnter two numbers: ”);

scanf (“%f %d”, &x, &y);

power = i = 1;

while (i <= y)

{

power = power \* x;

i++;

}

printf (“%f to the power %d is %f\n”, x, y, power);

return 0;

}

1. **Write down the simple interest program using for . Compare this program with the one that we wrote using while.**

/\* Calculation of simple interest for 3 sets of p, n and r \*/

**include**

int main()

{

int p, n, count;

float r, si;

for (count = 1; count <= 3; count = count + 1)

{

printf (“Enter values of p, n, and r ”);

scanf (“%d %d %f”, &p, &n, &r);

si = p \* n \* r / 100;

printf (“Simple Interest = Rs.%f\n”, si);

}

return 0;

}

1. **Write down the program to print numbers from 1 to 10 in different ways. This time we would use a for loop instead of a while loop.**

(a) # include

int main()

{

int i;

for (i = 1; i <= 10; i = i + 1)

printf (“%d\n”, i);

return 0;

}

.

(b) # include

int main()

{

int i;

for (i = 1; i <= 10;)

{

printf (“%d\n”, i);

i = i + 1;

}

return 0;

}

(c) # include

int main()

{

int i = 1;

for (; i <= 10; i = i + 1)

printf (“%d\n”, i);

return 0;

}

(d) # include

int main()

{

int i = 1;

for (; i <= 10;)

{

printf (“%d\n”, i);

i = i + 1;

}

return 0;

}

(e) # include

int main()

{

int i;

for (i = 0; i++ < 10;)

printf (“%d\n”, i);

return 0;

}

(f) # include

int main()

{

int i;

for (i = 0; ++i <= 10;)

printf (“%d\n”, i);

return 0;

}

1. **Write a program to determine whether a number is prime or not. A prime number is said to be prime if it is divisible only by 1 or itself.All we have to do to test whether a number is prime or not, is to divide it successively by all numbers from 2 to one less than itself. If remainder of any of these divisions is zero, the number is not a prime. If no division yields a zero then the number is a prime number.**

Include<stdio.h>

int main()

{

int num, i;

printf (“Enter a number ”);

scanf (“%d”, &num);

i = 2;

while (i <= num - 1)

{

if (num % i == 0)

{

printf (“Not a prime number\n”);

break;

}

i++;

}

if (i == num)

printf (“Prime number\n”);

}

.

1. **Write a program to print all prime numbers from 1 to 300.**

**Program**

/\* Generate all prime numbers from 1 to 300 \*/

**include**

int main()

{

int i, n = 1;

printf (“\nPrime numbers between 1 and 300 are :\n1\t”);

for (n = 1; n <= 300; n++)

{

i = 2;

for (i = 2; i < n; i++)

{

if (n % i == 0)

break;

}

if (i == n)

printf (“%d\t”, n);

}

return 0;

}

1. **Write a program to add first seven terms of the following series using a for loop.**

**Program**

/\* Sum of first seven terms of a series \*/

**include**

int main()

{

int i = 1, j;

float fact, sum = 0.0;

for (i = 1; i <= 7; i++)

{

fact = 1.0;

for (j = 1; j <= i; j++)

fact = fact \* j;

sum = sum + i / fact;

}

printf (“Sum of series = %f\n”, sum);

return 0;

}

Write a program to generate all combinations of 1, 2 and 3 using for loop.

Program



/\* Generate all possible combinations of 1 2 3 \*/

**include**

int main()

{

int i = 1, j = 1, k = 1;

for (i = 1; i <= 3; i++)

{

for (j = 1; j <= 3; j++)

{

for (k = 1; k <= 3; k++)

printf (“%d %d %d\n”, i, j, k);

}

}

return 0;

}