**TASK NO 1**

**NESTED FOR LOOP**

#include <iostream>

using namespace std;

int main()

{

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

{

for(int k=1;k<=i;k++)

{

cout<<"\* ";

}

cout<<endl;

}

return 0;

}

**NESTED WHILE LOOP**

#include <iostream>

using namespace std;

int main()

{

int i=1;

while(i<=7)

{

**int j=1;**

while(j<=i)

{

cout<<"\* ";

j++;

}

cout<<endl;

i++;

}

return 0;

}

**NESTED DO WHILE LOOP**

**#**include <iostream>

using namespace std;

int main()

{

int i=1;

do{

int j = 1;

do{

cout<<"\* ";

j++;

}while(j<=i);

cout<<endl;

i++;

}

while(i<=7);

return 0;

}

**TASK NO 2**

**NESTED FOR LOOP**

include <iostream>

using namespace std;

int main() {

for (int i = 0; i <= 5; i++) {

cout << "Multiplication Table of " << i << endl;

for (int j = 1; j <= 10; j++)

{

cout << i << " x " << j << " = " << i \* j << endl;

}

cout << endl;

}

return 0;

}

**NESTED WHILE LOOP:**

**#**include <iostream>

using namespace std;

int main() {

int i = 0;

while ( i <= 5) {

**cout << "Multiplication Table of " << i << endl;**

**int j = 1;**

**while(j<=10)**

{

cout << i << " x " << j << " = " << i \* j << endl;

j++;

}

cout << endl;

i++;

}

return 0;

}

**NESTED DO WHILE LOOP**

#include <iostream>

using namespace std;

int main() {

int i = 0;

do{

cout << "Multiplication Table of " << i << endl;

int j = 0;

do {

cout << i << " x " << j << " = " << i \* j << endl;

j++;

}while(j<=10);

cout << endl;

i++;

}while (i<=5);

return 0;

}

**TASK NO 3**

**NESTED FOR LOOP**

#include <iostream>

using namespace std;

int main()

{

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

{

for(int k=1;k<=7;k++){

cout<<"\* ";

}cout<<endl;

}

return 0;

}

**NESTED WHILE LOOP**

#include <iostream>

using namespace std;

int main()

{

int i=1;

while (i<=7)

{

int j=1;

while(j<=7){

cout<<"\* ";

j++;

}cout<<endl;

i++;

}

return 0;

}

**NESTED DO WHILE LOOP**

#include <iostream>

using namespace std;

int main()

{

int i=1;

do{

int j=1;

do {

cout<<"\* ";

j++;

}while (j<=7);

cout<<endl;

i++;

}while (i<=7);

return 0;

}

**TASK NO 4**

**NESTED FOR LOOP**

#include <iostream>

using namespace std;

int main() {

int num;

long long factorial = 1;

cout << "Enter a positive integer: ";

cin >> num;

for (int i = 1; i <= num; i++) {

for (int j = 1; j <= 1; j++) {

factorial \*= i;

}

}

cout << "Factorial of " << num << " = " << factorial << endl;

return 0;

}

**NESTED WHILE LOOP**

#include <iostream>

using namespace std;

int main() {

int num;

long factorial = 1;

cout << "Enter a positive integer: ";

cin >> num;

int i = 1;

while (i <= num) {

int j = 1;

while (j <= 1) {

factorial \*= i;

j++;

}

i++;

}

cout << "Factorial of " << num << " = " << factorial << endl;

return 0;

}

**NESTED DO WHILE TASK**

#include <iostream>

using namespace std;

int main() {

int num;

long long factorial = 1;

cout << "Enter a positive integer: ";

cin >> num;

int i = 1;

do {

int j = 1;

do {

factorial \*= i;

j++;

} while (j <= 1);

i++;

} while (i <= num);

cout << "Factorial of " << num << " = " << factorial << endl;

return 0;

}

**TASK NO 5**

**NESTED FOR LOOP**

#include <iostream>

using namespace std;

int main() {

int num, sum = 0;

cout << "Enter a positive integer: ";

cin >> num;

for (int i = num; i > 0; i /= 10) {

for (int j = 0; j < 1; j++) {

int digit = i % 10;

sum += digit;

}

}

cout << "Sum of the digits = " << sum << endl;

return 0;

}

**NESTED WHILE LOOP**

#include <iostream>

using namespace std;

int main()

{

int number;

int sum = 0;

cout << "Enter a number: ";

cin >> number;

while (number > 0)

{

int dig= number % 10;

while (dig> 0)

{

sum += dig;

break;

}

number /= 10;

}

cout << "The sum of the digits is: " << sum << "\n";

return 0;

}

**NESTED DO WHILE LOOP**

#include <iostream>

using namespace std;

int main() {

int num, sum = 0;

cout << "Enter a positive integer: ";

cin >> num;

int i = num;

do {

int j = 0;

do {

int digit = i % 10;

sum += digit;

j++;

} while (j < 1);

i /= 10;

} while (i > 0);

cout << "Sum of the digits = " << sum << endl;

return 0;}