

Section A: Operators (Write by Hand + Type in CS50.dev)

OP1. Input two integers and print sum, difference, product, quotient, and remainder.

```
#include <cs50.h>
#include <stdio.h>
int main(void)
{
    int a = get_int("Enter a");
    int b = get_int("Enter b");
    printf("\n SUM = %d", a+b);
    printf("\n DIFFERENCE = %d", a-b);
    printf("\n PRODUCT = %d", a*b);
    printf("\n QUOTIENT = %d", a/b);
    printf("\n REMAINDER = %d", a%b);
    return 0;
}
```

OP2. Convert temperature from Celsius to Fahrenheit.

```
#include <cs50.h>
#include <stdio.h>
int main(void)
{
    float c = get_int("Enter temperature in celsius");
    float f;
    f = ((c*9)/5)+32;
    printf("\n F = %f", f);
    return 0;
}
```

OP3. Check whether a number is even or odd.

```
#include <stdio.h>
#include <cs50.h>
int main(void)
{
    int x = get_int("Enter x");
    if (x%2 == 0)
    {
        printf("%d is even");
    }
    else
    {
        printf("%d is odd");
    }
    return 0;
}
```

OP4. Demonstrate increment and decrement operators.

```
#include <stdio.h>
#include <cs50.h>
int main (void)
{
    int x = 3;
    printf ("\\n increment %d", ++x);
    printf ("\\n Decrement %d", --x);
    return 0;
}
```

Section B: Conditions (Write by Hand + Type in CS50.dev)

COND1. Check whether a number is positive, negative, or zero.

```
#include <CS50.h>
#include <stdio.h>
int main(void)
{
    int x = get_int("Enter x:");
    if (x > 0)
    {
        printf("%d is positive");
    }
    else if (x == 0) { printf("%d Enter x is zero"); }
    else { printf("%d x is negative"); }
    return 0;
}
```

COND2. Print grade using else-if conditions.

```
#include <CS50.h>
#include <stdio.h>
int main(void)
{
    int grade = get_int("Enter grade:");
    if (grade >= 90 & grade <= 100)
        printf("A");
    else if (grade >= 60)
        printf("C");
    else
        printf("Fail");
    return 0;
}
```

COND3. Print the larger of two numbers or Equal.

```
#include <CS50.h>
#include <stdio.h>
int main(void)
{
    int p = get_int("Enter p:");
    int q = get_int("Enter q:");
    if (p > q)
        printf("p is greater");
    else if (p < q)
        printf("q is greater");
    else
        printf("p and q are equal");
    return 0;
}
```

COND4. Check whether a year is a leap year.

```
# include <stdio.h>
# include <cs50.h>
int main(void)
{
    int year = get_int("Enter a year:");
    if (year % 400 == 0)
    {
        printf("Leap Year");
    }
    else if (year % 4 == 0 && year % 100 != 0)
    {
        printf("Leap Year");
    }
    else
    {
        printf("Not a Leap Year");
    }
    return 0;
}
```

COND5. Nested if: Ticket Price Calculation

You are asked to write a C program that calculates the ticket price for a person based on their age and student status.

Input:

- Age of the person (integer)
- Student status (1 if the person is a student, 0 if not)

Ticket price rules:

- If age is less than 12, ticket price = 50
- If age is between 12 and 60 (inclusive):
 - If the person is a student, ticket price = 80
 - If the person is not a student, ticket price = 100
- If age is greater than 60, ticket price = 60

Output: Print the final ticket price.

Note: You must use a nested if structure to solve this problem.

```
# include <stdio.h>
# include <cs50.h>
int main(void)
{
    int age = get_int("Enter age");
    int s = get_int("Are you a student?");
    int price;
    if (age < 12) { price = 50; }
    else if (age <= 60) { if (s == 1) { price = 80; } else { price = 100; } }
    else { price = 60; }
    printf("Ticket Price: %d/h", price);
    return 0;
}
```

Section C: Loops (Write by Hand + Dry Run + Type in CS50.dev)

LOOP1. Print numbers from 1 to 10. Student Code (Write by Hand):

```
#include <cs50.h>
#include <stdio.h>
int main(void)
{
    int i;
    for (i=1; i<=10; i++)
    {
        printf ("%d\n", i);
    }
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	i = 1		Print 1
2	i = 2		Print 2
3	i = 3		Print 3
4	i = 4		Print 4
5	i = 5 to 10		Print 5 to 10, loop ends

LOOP2. Print numbers from 5 to 1. Student Code (Write by Hand):

```
#include <stdio.h>
#include <cs50.h>
int main(void)
{
    int i;
    for (i=5; i>=1; i--)
    {
        printf ("%d\n", i);
    }
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	$i = 5$		Print 5
2	$i = 4$		Print 4
3	$i = 3$		Print 3
4	$i = 2$		Print 2
5	$i = 1$		Print 1, loop ends

LOOP3. Input n . Print numbers from 1 to n . Student Code (Write by Hand):

```
#include <cs50.h>
#include <stdio.h>
int main(void)
{
    int n = get_int("Enter n:");
    for (int i = 1; i <= n; i++)
    {
        printf("%d\n", i);
    }
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	$i = 1$	$n = 5$	Print 1 or loop starts
2	$i = 2$	$n = 5$	Print 2
3	$i = 3$	$n = 5$	Print 3
4	$i = 4$	$n = 5$	Print 4
5	$i = 5 \rightarrow 6$	$n = 5$	Print 5; Condition fails

LOOP4. Input n . Print all even numbers from 1 to n . Student Code (Write by Hand):

```
#include <cs50.h>
#include <stdio.h>
int main(void)
{
    int n = get_int("Enter n:");
    int i;
    for (int i = 2; i <= n; i += 2) {
        printf("%d\n", i);
    }
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	i = 2	n = 10	Loop start, Print 2
2	i = 4	n = 10	Print 4
3	i = 6	n = 10	Print 6
4	i = 8	n = 10	Print 8
5	i = 10 -> 12	n = 10	Print 10, condition fails and loop ends.

LOOP5. Input n. Print all odd numbers from 1 to n. Student Code (Write by Hand):

```
#include <cs50.h>
#include <stdio.h>
int main(void)
{
    int n = get_int("Enter n");
    int i;
    for (i = 1; i <= n; i = i + 2)
    {
        printf("%d\n", i);
    }
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	i = 1	n = 10	Print = 1
2	i = 3	n = 10	Print = 3
3	i = 5	n = 10	Print = 5
4	i = 7	n = 10	Print = 7
5	i = 9 -> 11	n = 10	Print 9 condition fails

LOOP6. Find the sum of numbers from 1 to 10. Student Code (Write by Hand):

```
#include <cs50.h>
#include <stdio.h>
int main(void)
{
    int sum = 0;
    int i;
    for (i = 1; i <= 10; i++)
    {
        sum = sum + i;
    }
    printf("Sum = %d", sum);
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	i=1	Sum=0	Sum = 0 + 1 = 1
2	i=2	Sum=1	Sum = 1 + 2 = 3
3	i=3	Sum=3	Sum = 3 + 3 = 6
4	i=4	Sum=6	Sum = 6 + 4 = 10
5	i=5 to 10	Sum=10	Sum = 10 loop ends

LOOP7. Input n . Find the sum from 1 to n . Student Code (Write by Hand):

```
#include <cs50.h>
#include <stdio.h>
int main
{
    int n = get_int("Enter n");
    int sum = 0;
    int i;
    for (i=1; i<=n; i++)
    {
        sum = sum + i;
    }
    printf("sum = %d", sum);
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	i=1	n=5, sum=0	Sum = 0 + 1 = 1
2	2	Sum=1	1+2=3
3	3	Sum=3	3+3=6
4	4	Sum=6	6+4=10
5	5 to 6	Sum=10	10+5=15 loop ends

LOOP8. Print multiplication table of n . Student Code (Write by Hand):

```
#include <cs50.h>
#include <stdio.h>
int main(void)
{
    int n = get_int("Enter n");
    int i;
    for (i=1; i<=10; i++)
    {
        printf("%d * %d = %d\n", n, i, n*i);
    }
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	$i = 1$	3	$3 \times 1 = 3$
2	2	3	$3 \times 2 = 6$
3	3	3	$3 \times 3 = 9$
4	4	3	$3 \times 4 = 12$
5	5	3	$3 \times 5 = 15$ loop ends.

LOOP9. Factorial of a Number (Concept-Based)

Concept: The factorial of a non-negative integer n is written as $n!$ and means:

$$n! = 1 \times 2 \times 3 \times \dots \times n$$

Input: An integer n (assume $n \geq 0$)

Output: The factorial value of n

Example: Input: 5 Output: 120

Write a C program that uses a loop to compute the factorial.

Student Code (Write by Hand):

```
#include <stdio.h>
#include <cs50.h>
int main (void)
{
    int n = get_int("Enter n");
    int i;
    int factorial = 1;
    for (i = 1; i <= n; i++) { factorial *= factorial * i; }
    cout << "Factorial = " << factorial;
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	1	factorial	$1 \times 1 = 1$
2	2	1	$1 \times 2 = 2$
3	3	2	$2 \times 3 = 6$
4	4	6	$6 \times 4 = 24$
5	5	24	$24 \times 5 = 120$ loop ends.

LOOP10. Print first n multiples of 3. Student Code (Write by Hand):

```
#include <stdio.h>
#include <cs50.h>
int main(void)
{
    int n = get_int("Enter n");
    int a = 3;
    int b;
    int i;
    for (i = 1; i <= n; i++)
    {
        b = a * i;
        printf("%d * %d\n", a, i, b);
    }
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	1	a = 3, n = 5, b = 3	Print $3 \times 1 = 3$
2	2	b = 3 + 2	$3 + 2 = 6$
3	3	3×3	$3 \times 3 = 9$
4	4	3×4	$3 \times 4 = 12$
5	5	3×5	$3 \times 5 = 15$ loop ends.

LOOP11. Count numbers divisible by 5 from 1 to n . Student Code (Write by Hand):

```
#include <stdio.h>
#include <cs50.h>
int main(void)
{
    int n = get_int("Enter n");
    int i;
    int count = 0;
    for (i = 1; i <= n; i++) { if (i % 5 == 0)
    {
        count = count + 1;
    }
    printf("Count = %d\n", count);
}
return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	1	$1 \times 5 : 0$; count = 0	1 is not divisible by 5
2	2	$2 \times 5 : 0$; count = 0	2 is not divisible by 5
3	3	$3 \times 5 : 0$; count = 0	3 is not divisible by 5
4	4	$4 \times 5 : 0$; count = 0	4 is not in "S"
5	5	$5 \times 5 : 0$; count = 1	5 is divisible by 5 count = 1

LOOP12. Sum of first n even numbers. Student Code (Write by Hand):

```
# include <stdio.h>
# include <cs50.h>
int main (void)
{
    int n = get_int ("Enter n"),
        i,
        sum = 0;
    for (i = 1; i <= n; i++)
    {
        if (i % 2 == 0) { sum = sum + i; }
    }
    printf ("%d", sum);
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	i = 1	$1 \times 2 : 0$; sum = 0	1 is not even, sum = 0
2	2	$2 \times 2 : 0$; sum = 0+2	2 is even, sum = 2
3	3	$3 \times 2 : 0$; sum = 2	3 is not even, sum = 2
4	4	$4 \times 2 : 0$; sum = 2+4	4 is even, 2+4 = 6
5	5	$5 \times 2 : 0$; sum = 6	5 is not even, sum = 6

LOOP13. Sum of first n odd numbers. Student Code (Write by Hand):

```
# include <stdio.h>
# include <cs50.h>
int main (void)
{
    int n = get_int ("Enter n"),
        i,
        sum = 0;
    for (i = 1; i <= n, i++) { if (i % 2 != 0) { sum = sum + i; } }
    printf ("%d", sum);
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	1	$1 \mod 2 \neq 0; \text{sum} = 0$	1 is odd, sum = 1
2	2	$2 \mod 2 = 0; \text{sum} = 1$	2 is even, sum = 2
3	3	$3 \mod 2 \neq 0; \text{sum} = 1 + 3$	3 is odd, sum = 4
4	4	$4 \mod 2 = 0; \text{sum} = 4$	4 is even, sum = 4
5	5	$5 \mod 2 \neq 0; \text{sum} = 4 + 5 = 9$	5 is odd, sum = 9

LOOP14. Count digits in a number. Student Code (Write by Hand):

```
#include <stdio.h>
#include <cs50.h>
int main(void)
{
    int n = get_int("Enter n");
    int count = 0;
    int x = n;
    while (x != 0) { x = x / 10; count++ }
    printf("%d", count);
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	x = 324	0	x % 10 divided by 10, count = 1
2	32	1	x % 10 divided by 10, count = 2
3	3	2	x % 10 divided by 10, count = 3
4	0	3	x = 0, loop end
5	-	3	Printn count = 3

LOOP15. Star pattern:

```
*
**
***
****
```

Student Code (Write by Hand):

```
#include <stdio.h>
#include <cs50.h>
int main(void)
{
    int i;
    for (i = 1; i < 4; i++)
    {
        for (i = 1; i <= i; i++)
        {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	1	1	Print 1 star *
2	2	2	Print 2 star *
3	3	3	Print 3 star * * * * *
4	4	4	Print 4 star * * * * *
5	5	false	Loop end

LOOP16. Number pattern:

1
12
123
1234

Student Code (Write by Hand):

```
#include <stdio.h>
#include <cs50.h>
int main (void)
{int i,j;
for (i=1; i<=4; i++)
{
    for (j=1; j<=i; j++)
    {
        printf ("%d", j);
        printf ("\n");
    }
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	1	j=1	Print 1
2	2	j=2	Print 1 2
3	3	j=3	Print 1 2 3
4	4	j=4	Print 1 2 3 4
5	5	false	Loop end

LOOP17. Number pattern:

1
22
333
4444

Student Code (Write by Hand):

```
#include <stdio.h>
#include <cs50.h>
int main(void)
{
    int i, j;
    for (i = 1; i <= 4; i++)
    {
        for (j = 1; j <= i; j++)
        {
            printf("%d", j);
        }
        printf("\n");
    }
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	i = 1	j = 1	Print 1
2	2	2	Print 22
3	3	3	Print 333
4	4	4	Print 4444
5		false	Loop ends

LOOP18. Floyd's Triangle:

1
2 3
4 5 6
7 8 9 10

Student Code (Write by Hand):

```
#include <stdio.h>
#include <cs50.h>
int main(void)
{
    int i, j, num = 1;
    for (i = 1; i <= 4; i++)
    {
        for (j = 1; j <= i; j++)
        {
            printf("%d", num);
            num++;
        }
        printf("\n");
    }
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	i = 1	i = 1	Print 1
2	2	2	Print 2 3
3	3	3	Print 4 5 6
4	4	4	Print 7 8 9 10
5	5 false	false	loop ends

LOOP19. Reverse star pattern:

```
****
 ***
 **
 *
```

Student Code (Write by Hand):

```
#include <stdio.h>
#include <cs50.h>
int main(void)
{
    int i, j;
    for (i = 4; i >= 1; i--)
    {
        for (j = 1; j <= i; j++)
            printf("%*s", j);
        printf("\n");
    }
    return 0;
}
```

Dry Run Table (Mandatory):

Step	Loop variable(s)	Key variable(s)	Output / Explanation
1	i = 4	4	Print 4 return, ***
2	3	3	Print 3 return, ** *
3	2	2	Print 2 stars **
4	1	1	Print 1 star *
5	0	0, i >= 1	loop stops because i >= 2,

Final Note: You must type the code for each question in the CS50.dev environment. Handwritten code, dry run, and submitted code files must match.