

Instructions: Please write each program from scratch on your own. No copying and pasting from slides or use of tools like ChatGPT should be deployed. You must carefully look into the class materials before attempting and ask for help from your designated TA if you fail after a sincere effort.

You are discouraged from talking loudly in the lab, as others may feel disturbed. You may discuss with your neighboring batchmate, but in a soft tone. You are also discouraged from using the lab computers for other activities except solving the assignment during the lab session.

Set 1

1. Write a program in C programming language to print your name on the terminal.
2. Write a program in C programming language to take a number as input from the user and print its square on the terminal.
3. Write a program in C programming language to take as input two numbers and swap them using a third variable.
4. Write a program in C programming language to take as input two numbers and swap them without using a third variable.

```
#include<stdio.h>
// Program in C programming language to take
//as input two numbers and swap them without using a third variable

void main()
{
    int a, b;
    printf("Give two numbers as input:\n");
    scanf("%d %d", &a, &b); //a =5, b = 3
    printf("The numbers before swap ==> a = %d, b = %d\n\n", a, b);
    a = a + b; //a = 8, b = 3
    b = a - b; //a = 8, b = 5
    a = a - b; //a = 3, b = 5

    printf("The numbers after swap ==> a = %d, b = %d\n\n", a, b);
}
```

5. Write a program in C programming language to take as input the base (b) and height (h) of a triangle and print the area on the terminal (formula: $\text{area} = (\frac{1}{2}) * \text{base} * \text{height}$).
6. Write a program in C programming language to take the principal amount (P), time in years (T), and rate of interest (R) as input and print the Simple Interest (SI) and the Total Amount (A) on the terminal (formula: $\text{SI} = (\text{P} * \text{T} * \text{R}) / 100$, $\text{A} = \text{P} + \text{SI}$).

```
//C program to take the principal amount (P), time in years (T),  
//and rate of interest ( R ) as input  
//and print the Simple Interest (SI) and the Total Amount (A) on the terminal
```

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

```
void main()  
{  
    float P, T, R, SI, A;  
    printf("Give principal amount, time in years and rate of interest below:\n");  
    scanf("%f %f %f", &P, &T, &R);  
    SI = (P*T*R)/100;  
    A = P + SI;  
    printf("The Simple Interest is %f, the total amount is %f", SI, A);  
}
```

7. Write a program in C programming language to take as input the radius (R) and height (H) of a cone and print the volume on the terminal (formula: $\text{volume} = \text{PI} * \text{r} * \text{r} * (\text{h} / 3)$).
8. Write a C program to take the name (e.g., Kripa) and age (41) (supposing your name is "Sachin" and your age is 50) as input from the user and print a message on the terminal: "Hello Kripa, you are 41 years old. I am Sachin, I am 50 years old".

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

```
void main()  
{  
    char name[25];  
    int age;  
    printf("Give your name: ");  
    scanf("%s", name);  
    printf("\nGive your age: ");  
    scanf("%d", &age);
```

```
printf("Hello %s, you are %d years old. I am Sachin, I am 50 years old", name,
age);
}
```

9. Correct the following C programs and run the corrected versions:

i)

//Desired output: Hello world!\n

```
int main()
{
    print("Hello world!\n")
}
```

#include<stdio.h>

```
int main()
{
    printf("Hello world!\n");
}
```

ii)

//Make sure the program runs. The value of k is not important.

#include<stdio.h>

```
int main()
{
    int i = 5, j = 6, k;

    k = (i + j)/(j - 6);

    printf("k = %d\n", k);
}
```

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

```
int main()
```

```
{
```

```
    int i = 5, j = 6, k;
```

```
    k = (i + j)/(j - 5);
```

```
    printf("k = %d\n", k);
```

```
}
```

iii)

//Desired output: Miss mona please transfer Rs. 25000 to my account.

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

```
int main()
```

```
{
```

```
    int amount = 25,000;
```

```
    char name[] = 'mona';
```

```
    printf("Miss %s please transfer Rs. %d to my account.\n", name, amount);
```

```
}
```

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

```
int main()
```

```
{
```

```
    int amount = 25000;
```

```
    char name[] = "mona";
```

```
    printf("Miss %s please transfer Rs. %d to my account.\n", name, amount);
```

```
}
```

Set 2

1. Write a program in C to calculate the Gravitation Force (F) between two objects of masses M1 and M2 (Formula: $F = (G \cdot M1 \cdot M2) / (R \cdot R)$, $G = 6.673 \times 10^{-11}$, R is the distance between the two centres of masses of the two objects) such that G is i) a macro (#define), ii) local variable, iii) global variable.

```
#include<stdio.h>
#include<math.h>

//#define G 6.673*10E-11
float G = 6.673*10E-11;

void main()
{
    float F, M1 = 1000, M2 = 2000, R = 50;
    float G = 6.673*10E-11;

    F = (G*M1*M2)/(R*R);
    printf("F = %.20f\n", F);
}
```

2. Name functions as RollaCosta, Dominos, and Bawarchi – three restaurants in Kalyani. Every time you call a function, a message will be printed indicating the visit number of that function (restaurant). E.g., if you call RollaCosta, it will print “Welcome to Rolla Costa! This is your visit number 2” (assuming this is your 2nd call of the function). Implement this scheme using C programming language by using i) global variables and ii) static variables.

i) Global variables

```
#include<stdio.h>

int visit_rollacosta = 0;
int visit_dominos = 0;
int visit_bawarchi = 0;

void RollaCosta()
{
```

```

        visit_rollacosta++;
        printf("Welcome to Rolla Costa! This is your visit number %d\n\n",
visit_rollacosta);
    }

void Dominos()
{

        visit_dominos++;
        printf("Welcome to Dominos! This is your visit number %d\n\n",
visit_dominos);
    }

void Bawarchi()
{

        visit_bawarchi++;
        printf("Welcome to Bawarchi! This is your visit number %d\n\n",
visit_bawarchi);
    }

void main()
{
        RollaCosta();
        RollaCosta();
        RollaCosta();

        Dominos();
        Dominos();
        Dominos();

        Bawarchi();
        Bawarchi();
        Bawarchi();

    }

```

ii) Static variables

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

```

void RollaCosta()
{
    int static visit = 0;
    visit++;
    printf("Welcome to Rolla Costa! This is your visit number %d\n\n", visit);
}

void Dominos()
{
    int static visit = 0;
    visit++;
    printf("Welcome to Dominos! This is your visit number %d\n\n", visit);
}

void Bawarchi()
{
    int static visit = 0;
    visit++;
    printf("Welcome to Bawarchi! This is your visit number %d\n\n", visit);
}

void main()
{
    RollaCosta();
    RollaCosta();
    RollaCosta();

    Dominos();
    Dominos();
    Dominos();

    Bawarchi();
    Bawarchi();
    Bawarchi();

}

```

3. Consider a secret agent service with four secret agents: 002, 003, 005, and 007. The agents do not know each other, and they interact directly with the head of the organization (main()). The head interacts (function calls) with each agent separately, and in response, the agent prints his/her passcode (on the console).

The passcode of agent 00i ($i = 2, 3, 5, 7$) is an integer divisible by i . Right after the response, each agent updates the passcode such that it remains divisible by i (the updation scheme is known only to the agent and the head). There can be multiple such interactions. Note that the passcode of one agent is not visible to any other agent. The head can only view it upon the interaction (after which it is updated by the agent), but can not access it directly (the head knows that it is divisible by i for a sanity check, but does not know it exactly before the interaction). Using C and an appropriate storage class (variable type), implement this scheme. You may choose your own initial passcode for each agent i such that it is divisible by i .

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

```
void agent2()
```

```
{  
    static int passcode = 1234;  
  
    printf("Agent 002 reporting! Passcode = %d\n", passcode);  
    passcode += 2;  
}
```

```
void agent3()
```

```
{  
    static int passcode = 1233;  
  
    printf("Agent 003 reporting! Passcode = %d\n", passcode);  
    passcode += 3;  
}
```

```
void agent5()
```

```
{  
    static int passcode = 1235;  
  
    printf("Agent 005 reporting! Passcode = %d\n", passcode);  
    passcode += 5;  
}
```

```
void agent7()
```

```
{  
    static int passcode = 1239;  
  
    printf("Agent 007 reporting! Passcode = %d\n", passcode);  
    passcode += 7;  
}
```



```

void main()
{
    agent2();
    agent3();
    agent5();
    agent7();

    agent2();
    agent3();
    agent5();
    agent7();

    agent2();
    agent3();
    agent5();
    agent7();

    agent2();
    agent3();
    agent5();
    agent7();
}

```

4. Write a program to simulate the updation process of the current stock for three books 1, 2, and 3 handled by a librarian through three functions updateStockbook_i (i = 1, 2, 3) using a) global variables, b) static variables. The update is an increment of 1 in the stock of that book at a time. Print the current stock within the corresponding function after the update.

```

#include<stdio.h>

void updateStockbook_1()
{
    static int book1 = 0;

    book1++;

    printf("The current stock of book 1 is %d.\n", book1);
}

```

```

void updateStockbook_2()
{
    static int book2 = 0;

    book2++;

    printf("The current stock of book 2 is %d.\n", book2);

}

void updateStockbook_3()
{
    static int book3 = 0;

    book3++;

    printf("The current stock of book 3 is %d.\n", book3);

}

int main()
{
    updateStockbook_1();
    updateStockbook_2();
    updateStockbook_3();

    updateStockbook_1();
    updateStockbook_2();
    updateStockbook_3();

    updateStockbook_1();
    updateStockbook_2();
    updateStockbook_3();

}

```

5. Identify the issue in the following code and resolve it, ensuring none of the existing variables are removed and no line is commented out.

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

```
int main() {  
  
    int x = 10;  
  
    { //Block 1  
        int y = 20;  
        { //Block2  
            int x = 99;  
  
            printf("Sum1: %d\n", x + y);  
        }  
  
        printf("Sum2: %d\n", x + y);  
    }  
  
    printf("Sum3: %d\n", x + y);  
  
    return 0;  
}
```

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

```
int main() {  
  
    int x = 10;  
  
    { //Block 1  
        int y = 20;  
        { //Block2  
            int x = 99;  
  
            printf("Sum1: %d\n", x + y);  
        }  
  
        printf("Sum2: %d\n", x + y);  
    }  
  
    int y = 5;
```

```

printf("Sum3: %d\n", x + y);

return 0;
}

```

Set 3

1. Write a program to print the absolute value of an integer (I), which is the negation of the integer I if I is negative, else it is the same as I. Apply both if-else and the conditional operator (? :). There is no need to take user input.

```

#include <stdio.h>

int main() {
    int num = -5;
    int abs_value = num;;

    if(num < 0)
    {
        abs_value = -num;
    }

    printf("Absolute value of %d is %d\n", num, abs_value);
    return 0;
}

```

```

#include <stdio.h>

int main() {
    int num = -5;
    int abs_value = (num < 0) ? -num : num;
    printf("Absolute value of %d is %d\n", num, abs_value);
    return 0;
}

```

2. Write a program to print the floor and ceil of a real number **without** using ceil() and floor() functions of math.h (or any other in-built functions). E.g., for a number n = 55.5, floor(n) = 55.0 (i.e., the immediately **previous** integer) and ceil(n) = 56.0 i.e., the immediately **next** integer). There is no need to take user input.

```

#include<stdio.h>

void main()
{
    int c, f;
    float n = 55.00001, fract_part;

    f = (int)n;//floor
    c = n;//ceil

    fract_part = n - f;

    if(fract_part > 0)
    {
        c = n + 1;
    }

    printf("ceil(%f) = %d floor(%f) = %d\n", n, c, n, f);
}

```

3. Write a program to find the maximum and the second maximum of three integers (taken as input from the terminal) using if-else.

```

#include<stdio.h>

void main()
{
    int a = 30, b = 40, c = 5, max, max2;

    //max = a;

    if(a > b)//a > b
    {

        if(a > c)
        {
            max = a;
            if(b > c)

```

```

        {
            max2 = b;
        }
        else
        {
            max2 = c;
        }
    }
    else//a <= c; a > b
    {
        max = c;//c > a; a > b ==> c > b
        max2 = a;//a>b
    }
}
else//b > a
{
    if(b > c)
    {
        max = b;
        if(a > c)
        {
            max2 = a;
        }
        else
        {
            max2 = c;
        }
    }
    else//b <= c
    {
        max = c; //b > a; c > b ==> c > a
        max2 = b;
    }
}

printf("Max = %d 2nd max = %d\n", max, max2);
}

```

4. Write a program to calculate the electricity bill based on the units and the rates as given below, with an appropriate decision-making structure (e.g., if-else-if). The units (an

integer) will be taken as input from the user, and the total bill (using “%f”), based on the rate, in Rs. will be printed on the terminal.

Units	Rate (Rs. per unit)
First 25	4.89
Next 35	5.40
Next 40	6.41
Next 50	7.16
Next 50	7.33
Next 100	7.33
Above 300	8.92

- Write a program to assign teams (one for the Females and another for the Males) to students for the annual sports meet based on the criteria below, with an appropriate decision-making structure (e.g., if-else-if). You should ask for the Roll Number (an integer) and Gender (a character – M/F; declared as char and read using “%c”) as input from the terminal and print the Team Name for the student.

Roll number	Gender	Team name
1 - 25	Female	Mary Kom Smashers
26- 50	Female	Smriti Mandana Dashers
> 50	Female	Manu Bhaker Snipers
1 - 20	Male	Virat Kohli Challengers
21 - 40	Male	R Praggnanandhaa Combaters
> 40	Male	Neeraj Chopra Warriors

- Consider a Genie standing in front of the treasure door. The Genie asks you a puzzle, e.g., “The day before two days after the day before tomorrow is Saturday. What day is it today?” with four options like a) Sunday, b) Monday..... The Genie lets you in only if the answer is correct; else imprisons you. Implement this scheme using switch-case, where you print messages based on success, failure, or invalid answer.

Can you also do it using a maximum of three “break”s?

```
#include<stdio.h>

void main()
{
    char option;

    printf("Give the correct option for the puzzle:\n");
    printf("Puzzle: The day before two days after the day before tomorrow is
Saturday. What day is it today?\n\n");
    printf("Options: a) Sunday, b) Monday, c) Friday, d) Thursday.\n\n");
    printf("Please input one of the above four options below:\n");
    scanf("%c", &option);

    switch(option)
    {
        case 'c': printf("Correct! The treasure is yours!\n");
                    break;

        case 'a':

        case 'b':

        case 'd': printf("Wrong! You are imprisoned!\n");
                    break;

        default: printf("Wrong option!!!");
                  break;
    }
}
```

7. An integer n when divided by another integer d (>0) belongs to exactly one of the mod d equivalence classes $(Z_d) \{[0], [1], [2], \dots [d-1]\}$ where $[i]$ (i being an integer) is the set of all integers with remainder i when divided by d such that $i = 0, 1, 2, \dots, d - 1$. That is, if $n = 20$, $d = 5$, n will belong to the class $[0]$; if $n = 21$, $d = 5$, n will belong to $[1]$, and so on. Design a C program to print the mod d equivalent class of n using switch case. Note that it doesn't need to be a generic solution for any d (no need to take 'n' and 'd' as user inputs, and you can have fixed n, d values).


```

#include<stdio.h>

void main()
{
    int n = 21, d = 5;

    switch(n%d)
    {
        case 0:    printf("[0]\n");
                   break;

        case 1:    printf("[1]\n");
                   break;

        case 2:    printf("[2]\n");
                   break;

        case 3:    printf("[3]\n");
                   break;

        case 4:    printf("[4]\n");
                   break;

    }
}

```

8. Write a C program to simulate the reception of an online e-shop. First, greet the customer by printing a welcome message. Then, print the catalog of five items (with a product id like 0, 1,..) with the prices and quantities present for each. Next, request the customer to provide the id of the product s/he want to buy. Using switch-case (that will move to the appropriate 'case' for the product), for each product, print a message asking for the quantity of the product selected (exit with an appropriate message if a wrong option has been provided for the product id). If the quantity is less than or equal to that of the available item, print the message with the final bill and update the stock (if the quantity requested is more than that available, print an appropriate message and exit). Finally, print a nice goodbye message.

```

#include<stdio.h>

void main()
{
    int apple_n = 5, mango_n = 10, banana_n = 50, pineapple_n = 40, papaya_n =
15;
    float apple_p = 10.0, mango_p = 50.0, banana_p = 5.0, pineapple_p = 55.5,
papaya_p = 30.5;
    int code, qty;

    printf("Welcome to Fruitopia!!\n\n");
    printf("Catalogue:\n\n");
    printf("Apple (code 0): quantity = %d price = %f INR per piece\n", apple_n,
apple_p);
    printf("Mango (code 1): quantity = %d price = %f INR per piece\n", mango_n,
mango_p);
    printf("Banana (code 2): quantity = %d price = %f INR per piece\n", banana_n,
banana_p);
    printf("Pineapple (code 3): quantity = %d price = %f INR per piece\n",
pineapple_n, pineapple_p);
    printf("Papaya (code 4): quantity = %d price = %f INR per piece\n", papaya_n,
papaya_p);

    printf("Please give the code of the fruit you want to buy!\n\n");
    scanf("%d", &code);

    switch(code)
    {
        case 0: //Apple
            printf("Quantity please:\n");
            scanf("%d", &qty);
            if(qty > apple_n)
            {
                printf("Sorry! We are out of stock!!\n");
            }
            else
            {
                printf("Bill = %f INR", qty*apple_p);
            }
            apple_n -= qty;
            break;

        case 1: //Mango
            printf("Quantity please:\n");

```

```
scanf("%d", &qty);
if(qty > mango_n)
{
    printf("Sorry! We are out of stock!!\n");
}
else
{
    printf("Bill = %f INR", qty*mango_p);
}
mango_n -= qty;
break;
```

case 2: //Banana

```
printf("Quantity please:\n");
scanf("%d", &qty);
if(qty > banana_n)
{
    printf("Sorry! We are out of stock!!\n");
}
else
{
    printf("Bill = %f INR", qty*banana_p);
}

banana_n -= qty;
break;
```

case 3: //Pineapple

```
printf("Quantity please:\n");
scanf("%d", &qty);
if(qty > pineapple_n)
{
    printf("Sorry! We are out of stock!!\n");
}
else
{
    printf("Bill = %f INR", qty*pineapple_p);
}
pineapple_n -= qty;
break;
```

case 4: //Papaya

```
printf("Quantity please:\n");
scanf("%d", &qty);
```

```

        if(qty > papaya_n)
        {
            printf("Sorry! We are out of stock!!\n");
        }
        else
        {
            printf("Bill = %f INR", qty*papaya_p);
        }
        papaya_n -= qty;
        break;

    default: printf("Invalid choice!");
            break;

}

printf("\nPlease visit again!!\n");

}

```

9. In Computer Science, peer-reviewed international conferences are widely considered to be highly prestigious, and they have a ranking as CORE A* (the highest), A, B, and C (see <https://portal.core.edu.au/conf-ranks/>). As an example, EMNLP is A*, shown below.

ICORE Conference Portal

CORE rankings is now an international collaboration named ICORE

Any rankings questions should be sent to icore.rankings@gmail.com

[ICORE rankings page](#) | [Frequently asked questions](#) | [Description of Conference Ranks](#) | [Fields of Research codes](#)

[CORE Dora discussion document](#) | [Guidelines for reporting rankings](#) | [Usage of CORE rankings](#) | [ICORE addition and upgrade requirements](#) | [Data used in Rankings a](#)

Search by: All

Source: CORE2023

Showing results 1 - 1 of 1

Title	Acronym	Source	Rank
Empirical Methods in Natural Language Processing	EMNLP	CORE2023	A*

Corporate R&D bodies provide incentives to employees if they have papers accepted at such venues, as that often boosts the global reputation of the corporate body. Implement a scheme to decide on the Incentive given the Conference Rank using a) if-else-if, b) switch as below:

Conference Rank	Incentive
A*	500 USD + funds for conference travel, accommodation, and registration
A	300 USD + funds for conference travel and accommodation
B	150 USD + funds for conference travel
C	100 USD only

Can you also modify your switch scheme (hint: adjusting 'break') if the company has the same incentive for both A* and A?

Also, note that "A*" can not be expressed as a character. So, you need to have a number or character mapping of the conference ranks.