

Name: Keshavulu Siddharth V.
Roll No: 25BCL119

C Programming Ass.- 2

①

* Section - I

I) Matrix Addition, Subtraction & Multiplication :-

```
#include <stdio.h>
int main() {
    int r,c;
    printf("Enter rows & cols.: ");
    scanf("%d %d", &r, &c);
    int A[r][c], B[r][c], C[r][c];
    printf("Enter Matrix A:\n");
    for (int i=0; i<r; i++) {
        for (int j=0; j<c; j++) {
            scanf("%d", &A[i][j]);
        }
    }
    printf("Enter Matrix B:\n");
    for (int i=0; i<r; i++) {
        for (int j=0; j<c; j++) {
            scanf("%d", &B[i][j]);
        }
    }
    for (int i=0; i<r; i++) {
        for (int j=0; j<c; j++) {
            C[i][j] = A[i][j] + B[i][j];
        }
    }
    printf("\nAddition:\n");
    for (int i=0; i<r; i++) {
        for (int j=0; j<c; j++) {
            printf("%d ", C[i][j]);
        }
        printf("\n");
    }
}
```

for (int i=0; i<r; i++)
 for (int j=0; j<c; j++) {
 C[i][j] = A[i][j] - B[i][j];
 }

printf ("\nSubtraction:\n");
for (int i=0; i<r; i++) {
 for (int j=0; j<c; j++)
 print ("%d", C[i][j]);
 printf ("\n");

int M[2][c];
for (int i=0; i<r; i++)
 for (int j=0; j<c; j++) {
 M[i][j] = 0;
 for (int k=0; k<c; k++)
 M[i][j] += A[i][k] * B[k][j];
 }

printf ("\nMultiplication:\n");
for (int i=0; i<r; i++) {
 for (int j=0; j<c; j++)
 print ("%d", M[i][j]);
 printf ("\n");

(9)

2) Sort 4x4 Matrix into 1D Array:-

```
#include <stdio.h>
int main(){
    int a[4][4], arr[16], k=0;
    printf("Enter 4x4 Matrix:\n");
    for (int i=0; i<4; i++){
        for (int j=0; j<4; j++){
            scanf("%d", &a[i][j]);
            arr[k] = a[i][j];
        }
    }
    for (int i=0; i<4; i++){
        for (int j=0; j<4; j++){
            if (arr[j] > arr[j+1]){
                int t = arr[j]; arr[j] = arr[j+1]; arr[j+1] = t;
            }
        }
    }
}
```

```
printf("Sorted array:\n");
for (int i=0; i<16; i++) printf("%d", arr[i]);
}
```

→ Output :- Enter 4x4 matrix: → Sorted array:

9 4 3 2	1 2 3 4 5 6 7 8 9 10 11 12
1 10 11 12	13 14 15 16
5 13 7 8	
14 6 15 16	

3) Largest & Smallest in 3×3 Matrix using pointer.

```
#include <stdio.h>
int main {
    int a[3][3], p = *a[0][0];
    printf("Enter 3x3 matrix:\n");
    for (int i=0; i<9; j++) {
        scanf("%d", p+i);
    }
    int max = *p, min = *p;
    for (int i=1; i<9; i++) {
        if (*p+i > max) max = *(p+i);
        if (*p+i < min) min = *(p+i);
    }
    printf("Largest = %d\n Smallest = %d\n", max, min);
}
```

→ Output

Enter 3x3 matrix:

1 9 8

5 2 4

7 6 3

Largest = 9

Smallest = 1

③

4) Accept & Print 3 Books Names (Array to Pointers)

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(){
    char *books[3], temp[100];
    printf("Enter 3 books:\n");
    for(int i=0; i<3; i++){
        scanf("%[^\\n]", temp);
        books[i] = malloc(strlen(temp)+1);
        strcpy(books[i], temp);
    }
    printf("\nBooks:\n");
    for(int i=0; i<3; i++)
        printf("%s\n", books[i]);
    free(books[0]);
}
```

→ Output:

Enter 3 books: → Books:
Computer Handling
Maths
English.

5) Name Abbreviation Program:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
int main () {
    char name [200], *parts [20];
    int count = 0;
    printf ("Enter full name:");
    gets (name, sizeof (name), stdin);
    name = strtok (name, "\n") = '0';
    char *token = strtok (name, " ");
    while (token) {
        parts [count] = token;
        token = strtok (NULL, " ");
        count++;
    }
    for (int i = 0; i < count - 1; i++)
        printf ("%c.", toupper (parts [i][0]));
    printf ("%c.%s", parts [count - 1]);
}
```

→ Output :-

Enter full name: Siddharth Keshwala

S. Keshwala

* Section - 2

2) Power (a^b)

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

```
int power(int a, int b){
```

```
    int result = 1;
```

```
    for (int i = 1; i <= b; i++)
```

```
        result *= a;
```

```
    return result;
```

```
}
```

```
int main(){
```

```
    int a, b;
```

```
    printf("Base: "); C("Base and exponent: ");
```

```
    scanf("%d %d", &a, &b);
```

```
    printf("%d ^ %d = %d", a, b, power(a, b));
```

```
}
```

->

Output:-

~~Enter~~ Base and exponent: 2 2

$2^2 = 4$.

2) Leap Year:

```
#include <stdio.h>
int isleap (int y) {
    if ((y % 4 == 0 && y % 100 != 0) || (y % 400 == 0))
        return 1;
    else
        return 0;
}

int main () {
    int year;
    printf ("Enter year:");
    scanf ("%d", &year);
    if (isleap (year)) {
        printf ("%d is leap year", year);
    }
    else
        printf ("%d is not leap year", year);
}
```

Output:

```
Enter year: 2025
2025 is not leap year.
```

3) Recursive Factorial Function.

```
#include <stdio.h>
int fact(int n) {
    if (n<=1) return 1;
    return n * fact(n-1);
}

int main() {
    int n;
    printf("Enter number: ");
    scanf("%d", &n);
    printf("Factorial = %d", fact(n));
}
```

→ Output;

Enter number: 5

Factorial = 120.

↳ Swap using call by value.

```
#include <stdio.h>
void swap (int x, int y) {
    int temp = x;
    x = y;
    y = temp;
    printf ("Inside Function : x=%d, y=%d\n", x, y);
}
```

```
int main () {
    int a = 10, b = 20;
    printf ("Before call : a=%d, b=%d\n", a, b);
    swap (a, b);
    printf ("After call : a=%d, b=%d\n", a, b);
}
```

→ Output.

Before call : a = 10, b = 20

~~After call~~ Inside function : x = 20, y = 10

After call : a = 10, b = 20

(5)

5) Find Min and Max using call by reference

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

```
void minmax(int a[], int n, int *min, int *max) {
    *min = *max = a[0];
    for (int i = 1; i < n; i++) {
        if (a[i] > *max) *max = a[i];
        if (a[i] < *min) *min = a[i];
    }
}
```

```
int main() {
```

```
    int n;
```

```
    printf("Enter array size: ");
    scanf("%d", &n);
```

```
    int a[n], min, max;
```

```
    printf("Enter %d Elements: ", n);
```

```
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);
```

```
    minmax(a, n, &min, &max);
```

```
    printf("Minimum=%d Maximum=%d", min, max);
```

```
}
```

→ Output:-

Enter array size: 3

Enter 3 Elements: 1 50 25

Minimum = 1

Maximum = 50

6) Calculator using Functions

```
#include <stdio.h>
int add (int a, int b) {return a+b;}
int sub (int a, int b) {return a-b;}
int mult (int a, int b) {return a*b;}
float div (int a, int b) {return (float)a/b;}
int main () {
    int a, b, ch;
    printf ("Enter two numbers: ");
    scanf ("%d%d", &a, &b);
    printf ("1. Add 2. Subtract 3. Multiply 4. Divide\n Enter choice: ");
    scanf ("%d", &ch);
    switch (ch) {
        case 1: printf ("Result = %d", add (a, b)); break;
        case 2: printf ("Result = %d", sub (a, b)); break;
        case 3: printf ("Result = %d", mult (a, b)); break;
        case 4: printf ("Result = %.2f", div (a, b)); break;
        default: printf ("Invalid choice");
    }
}
```

→ Output :- Enter two numbers : 10 10
1. Add 2. Subtract 3. Multiply 4. Divide
Enter choice : 3
Result = 100.

(7)

7) To Run all of above

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

```
int power (int a, int b){  
    int n = 1; for (int i=1; i<=b; i++) n *= a; return n;  
}
```

```
int isLeap (int y) { return ((y%4==0) && (y%100!=0)) || (y%400  
==0); }
```

```
int fact (int n) { return (n<=1) ? 1 : n * fact (n-1); }
```

```
void swap (int x, int y){
```

```
    int t=x; x=y; y=t;
```

```
    printf ("Inside : x=%d y=%d\n", x, y);
```

```
}
```

```
Void find Min Max (int a[], int n, int * min, int * max){
```

```
* min = * max = a[0];
```

```
for (int i=1; i<n; i++) {
```

```
    if (a[i] > * max) * max = a[i];
```

```
    if (a[i] < * min) * min = a[i];
```

```
}
```

```
}
```

```
int main(){
```

```
    int ch;
```

```
printf ("1. Power\n2. Leap\n3. Factorial\n4. Swap\n5. Min/Max\n6. Choice:\n");
```

```
scanf ("%d", &ch);
```

if (ch==1) {

```
int a,b; printf ("Enter a,b:"); scanf ("%d%d", &a, &b);
printf ("%d\n", power(a,b));
```

}

else if (ch==2) {

```
int y; printf ("Enter years:"); scanf ("%d", &y);
printf ("Is leap? %s\n", leap(y)?"Leap\n": "Not leap\n");
```

}

else if (ch==3) {

```
int n; printf ("Enter n:"); scanf ("%d", &n);
printf ("%d\n", fact(n));
```

}

else if (ch==4) {

```
int a=5, b=10;
```

```
printf ("Before: %d %d\n", a, b); swap(&a, &b);
```

```
printf ("After: %d %d\n", a, b);
```

}

else if (ch==5) {

```
int n, min, max; printf ("Enter n:"); scanf ("%d", &n);
int a[n]; for (int i=0; i<n; i++) scanf ("%d", &a[i]);
findMinMax(a, n, &min, &max);
printf ("Min = %d Max = %d\n", min, max);
```

}

else printf ("Invalid option");

.

(8)

8) Loop programs using Recursion Examples:

```
#include <stdio.h>
void printNum (int n) {
    if (n==0) return;
    printNum (n-1);
    printf ("%d", n);
}
```

 \rightarrow Output

Enter n: 3

1 2 3

```
int main() {
    int n;
    printf ("Enter n:");
    scanf ("%d", &n);
    printNum(n);
}
```

\rightarrow # include <stdio.h>

 \rightarrow Output:-

```
int sum (int n) {
    if (n==0) return 0;
    return n + sum (n-1);
}
```

Enter n: 3

1 2 3

Sum = 6.

```
int main() {
    int n;
    printf ("Enter n:");
    scanf ("%d", &n);
    printf ("Sum=%d", sum(n));
}
```