* ***Purpose :*** **Classwork**

* ***Date :*** **23/12/2025**

* ***Author :*** **Vikas Srivastava**

* ***ID :*** **55984**

* ***Batch ID :*** **25SUB4505**

---

1. ***Code :*** Write a program to demonstrate initialization and traversal of a 2-D array, where only the first column is initialized and the remaining elements are automatically set to zero.

arr2DOne.c

```c
#include <stdio.h>

int main(){
  int arr[][3] = {{10}, {20}, {30}, {40}, {50}};
  int row=5, col = 3;
  int rCnt, cCnt;
  for (rCnt=0; rCnt < row; rCnt++){
    for (cCnt=0; cCnt < col; cCnt++)
      printf("%d  ", arr[rCnt][cCnt]);
    printf("\n");
  }
}
```

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc arr2DOne.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
10  0  0
20  0  0
30  0  0
40  0  0
50  0  0
```

2. ***Code :*** Write a program to demonstrate how a 2-D array is stored in contiguous memory and how accessing elements beyond a row using column indexing still refers to valid memory locations (row-major order), resulting in undefined but observable values.

```c
#include <stdio.h>

int main(){
  int arr[][3] = {{10}, {20}, {30}, {40}, {50}};
  printf("arr[0][0] --> %d\n", arr[0][0]);
  printf("arr[0][3] --> %d\n", arr[0][3]);
  printf("arr[0][6] --> %d\n", arr[0][6]);
  printf("arr[0][9] --> %d\n", arr[0][9]);
  printf("arr[0][12] --> %d\n", arr[0][12]);
}
```

Output :

```
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc arr2DTwo.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
arr[0][0] --> 10
arr[0][3] --> 20
arr[0][6] --> 30
arr[0][9] --> 40
arr[0][12] --> 50
```

3. ***Code :*** Write a program to initialize an array with 100 consecutive numbers starting from a user-given value, use a separate function to check whether a number is prime, and replace all non-prime numbers in the array with 0.

```c
#include <stdio.h>
int isPrime(int);
int main(){
  int arr[100];
  printf("Enter the first Num: ");
  scanf("%d", &arr[0]);

  for (int cnt = 1; cnt <  100; cnt++)
    arr[cnt] = arr[0] + cnt; //assigning values

  printf("Arr: ");
  for (int cnt = 0; cnt <  100; cnt++) //printing
    printf("%d  ", arr[cnt]);
  printf("\n");


  for (int cnt = 0; cnt <  100; cnt++)
    if (isPrime(arr[cnt]) == 0)
      arr[cnt] = 0;

  printf("Arr: ");
  for (int cnt = 0; cnt <  100; cnt++) //printing
    printf("%d  ", arr[cnt]);
  printf("\n");
}
int isPrime(int num){
  int flag = 1;
  for (int cnt = 2; cnt < num; cnt++){
    if (num % cnt == 0){
      flag = 0;
      break;
    }
  }
  return flag;
}
```

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc arrTwo.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
Enter the first Num: 1
Arr: 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23
1  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62
  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100
Arr: 1  2  3  0  5  0  7  0  0  0  11  0  13  0  0  0  17  0  19  0  0  0  23  0  0  0
0  0  0  0  0  53  0  0  0  0  0  59  0  61  0  0  0  0  0  67  0  0  0  71  0  73  0
7  0  0  0
```

4. ***Code :*** Write a program to display a formatted monthly calendar by printing the days of the week and aligning dates correctly using loops, spacing, and modulo arithmetic based on the starting weekday.

dayOfWeekV1.c

```c

#include <stdio.h>
int main(){
  int w = 1;
  int cnt;
  printf("Su Mo Tu We Th Fr Sa\n");
  for (cnt = 0; cnt < w; cnt++)
    printf("%3s"," ");
  for (cnt = 1; cnt <= 30 ; cnt++){
    printf("%-3d", cnt);
    if ((w + cnt) % 7 == 0)
      printf("\n");
  }
  printf("\n");
}

```

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc dayOfWeekV1.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
Su Mo Tu We Th Fr Sa
   1  2  3  4  5  6
7  8  9  10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
```

5. ***Code :*** Write a program to demonstrate how multiple user-defined functions are declared, defined, and called from the main() function, showing the sequence of execution flow between main() and other functions.

funFlowOne.c

```c
#include <stdio.h>

void fun();
void funOne();
void funTwo();
void funThree();

int main(){
  printf("1.  in main()...\n");
  fun();
  funOne();
  funThree();
  printf("2.  in main()...\n");
}

void fun(){
  printf("Inside fun()...\n");
}
void funOne(){
  printf("Inside funOne()...\n");
}
void funTwo(){
  printf("Inside funTwo()...\n");
}
void funThree(){
  printf("Inside funThree()...\n");
}

```

Output :

```
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc funFlowOne.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
1.  in main()...
Inside fun()...
Inside funOne()...
Inside funThree()...
2.  in main()...
```

6. Code : Write a program to demonstrate recursion in C, where a function repeatedly calls itself to print numbers in ascending order during the recursive calls and in descending order while returning, illustrating the call stack behaviour.

funFlowThree.c

```c
#include <stdio.h>

void funOne(int);

int main(){
  funOne(1);
  printf("\n");
}

void funOne(int num){
  if (num <= 5){
    printf("%d  ", num);
    funOne(num + 1);
    printf("%d  ", num);
  }
}
```

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc funFlowThree.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
 1  2  3  4  5  5  4  3  2  1
```

7. ***Code :*** Write a program to demonstrate nested function calls in C, showing how control flows from main() to multiple functions and then returns back in Last-In–First-Out (LIFO) order.

funFlowTwo.c

```c
#include <stdio.h>
void fun();
void funOne();
void funTwo();
void funThree();
int main(){
  printf("1.  in main()...\n");
  fun();
  printf("2.  in main()...\n");
}
void fun(){
  printf("1. Inside fun()...\n");
  funOne();
  printf("2. Inside fun()...\n");
}
void funOne(){
  printf("1. Inside funOne()...\n");
  funTwo();
  printf("2. Inside funOne()...\n");
}
void funTwo(){
  printf("1. Inside funTwo()...\n");
  funThree();
  printf("2. Inside funTwo()...\n");
}
void funThree(){
  printf("Inside funThree()...\n");
}
```

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc funFlowTwo.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
1.  in main()...
1. Inside fun()...
1. Inside funOne()...
1. Inside funTwo()...
Inside funThree()...
2. Inside funTwo()...
2. Inside funOne()...
2. Inside fun()...
2.  in main()...
```

8. ***Code*** : Write a program to demonstrate the use of a function prototype (declaration), followed by function definition and function call, showing how a user-defined function is invoked from the main() function.

funFlowTwo.c | *funOne.c*

```c
#include <stdio.h>

//function declaration / prototype
void funOne(); // It a function taking no arguments and returning nothing


int main(){
  printf("Inside main()\n");
  funOne();//function calling
}

void funOne(){//function definition
  printf("Inside funOne...\n");
}
```

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc funOne.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
Inside main()
Inside funOne...
```

9. ***Code :*** Write a program to implement a user-defined function isPrime() that checks whether a given number is prime, and demonstrate calling this function directly inside printf() to display the result.

```c
#include <stdio.h>
//implementing isPrime(num)
int isPrime(int); //prototype
int main(){
  printf("Res: %d\n", isPrime(101));//function calling inside printf()
  printf("Res: %d\n", isPrime(100));//function calling inside printf()
}//ending
int isPrime(int num){ //definition
  int flag = 1;
  for (int cnt = 2; cnt < num; cnt++){
    if (num % cnt == 0){
      flag = 0;
      break;
    }
  }
  return flag;
}
```

Output :

```
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc isPrime.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
Res: 1
Res: 0
```

10. ***Code :*** Write a program to implement a user-defined string copy function similar to strcpy(), which copies characters from a source string to a destination string until the null character is encountered, and returns the destination string.

```c
#include <stdio.h>
#include <string.h>
char *myStrCpy(char *, const char *);

int main(){
  char str[25] = "Hello How0 I am fine";
  char strCopy[30];

  printf("str: %s\t\tLen: %lu\n",str, strlen(str));
  myStrCpy(strCopy, str);

  printf("Copied: %s\t\tLen: %lu\n", strCopy, strlen(strCopy));
}
char *myStrCpy(char *dest, const char *src){
  int cnt = 0;
  while(dest[cnt] = src[cnt])
    cnt++;
  return dest;
}
```

Output :

```
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc myStrCpy.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
str: Hello How0 I am fine               Len: 20
Copied: Hello How0 I am fine            Len: 20
```

11. ***Code*** : Write a program to implement a user-defined string length function similar to strlen(), which counts the number of characters in a string until the null character ('\0') is encountered and displays the string length.

myStrCpy.c | myStrLen.c

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

int myStrLen(const char *);

int main(){
  char str[25] = "Hello How0 I am fine";

  printf("str: %s\t\tLen: %d\t\tSize: %lu\n",str, myStrLen(str), sizeof(str));
}

int myStrLen(const char *str){
  int cnt = 0;
  while(str[cnt] != 0)
    cnt++;

  return cnt;
}
```

Output :

```
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc myStrLen.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
str: Hello How0 I am fine               Len: 20         Size: 25
```

12. ***Code*** : Write a program to demonstrate the use of nested for loops to print a sequential number pattern in a 5×5 matrix format, where numbers increase continuously across rows and columns.

myStrCpy.c | nestingLoopOne.c

```c
#include <stdio.h>

int main(){
  int cntOne, cntTwo, num = 1;
  for (cntOne=0; cntOne<5; cntOne++){
    for (cntTwo=0; cntTwo<5; cntTwo++)
      printf("%d ", num++);
    printf("\n");
  }
}
```

Output :

```
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc nestingLoopOne.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
```

13. ***Code*** : Write a program to demonstrate recursion by printing numbers from 1 to 10, where a function repeatedly calls itself until a base condition is met.

myStrCpy.c | recurOne.c

```c
#include <stdio.h>

void recur(int);

int main(){
  recur(1);
  printf("\n");
}

void recur(int num){
  if (num <= 10){
    printf("%d  ", num);
    recur(num + 1);
  }
}
```

Output :

```
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc recurOne.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
1  2  3  4  5  6  7  8  9  10
```

14. Code : Write a program to demonstrate the use of the strcmp() function to compare two strings, showing that it returns 0 for equal strings, a negative value when the first string is lexicographically smaller, and a positive value when it is greater.

myStrCpy.c | strFunFour.c

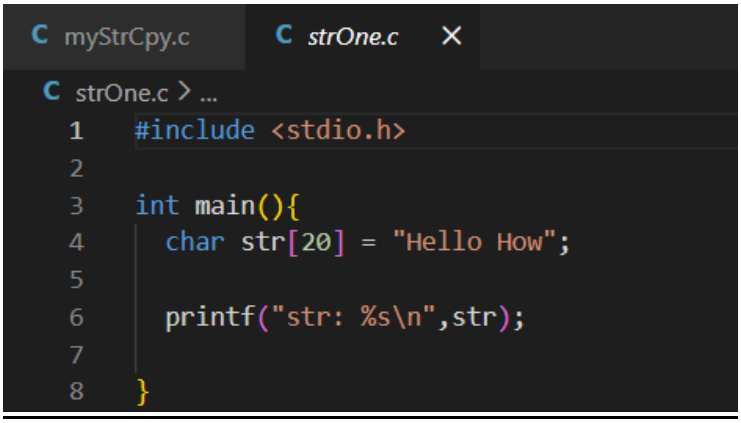
```c
#include <stdio.h>
#include <string.h>

int main(){
  printf("Compare : %d\n", strcmp("Hello","Hello")); //same string returns 0
  printf("Compare : %d\n", strcmp("Hello","hello")); //difference returns -1
  printf("Compare : %d\n", strcmp("hello","Hello")); //difference returns  1
}
```

Output :

```
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> gcc strFunFour.c
PS C:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C_CPP\Day_2\Classwork> .\a.exe
Compare : 0
Compare : -1
Compare : 1
```

15. ***Code :*** Write a program to demonstrate how a string is stored in a character array and printed on the screen using the %s format specifier in C.



Output :