

Assignment no. 1

C Program for

1. To Print factorial of the number
2. Print sum of digit in array of 10
3. Matrix addition of 3x3
4. Sum of array of 10 sum of odd no. and even no.
5. Size of all data types

1.

Ans.

```
#include <stdio.h>

unsigned long long factorial(int n) {
    if (n == 0 || n == 1)
        return 1;
    else
        return n * factorial(n - 1);
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    if(num < 0) {
        printf("Please enter a non-negative number.\n");
    }
    else {
        printf("Factorial of %d = %llu\n", num, factorial(num));
    }
    return 0;
}
```

Output :




bin.c	  	Output
<pre>#include <stdio.h> unsigned long long factorial(int n) { if (n == 0 n == 1) return 1; else return n * factorial(n - 1); } int main() { int num; printf("Enter a number: "); scanf("%d", &num); if(num < 0) { printf("Please enter a non-negative number.\n"); } else { printf("Factorial of %d = %llu\n", num, factorial(num)); } return 0; }</pre>		<pre>/tmp/Wwixho0U4.o Enter a number: 5 Factorial of 5 = 120 </pre>

2.

Ans.

```
#include <stdio.h>
int main() {
    int a[10], i;
    int sum = 0;
    printf("Enter the numbers in the array:\n");
    for (i = 0; i < 10; i++) {
        scanf("%d", &a[i]);
    }
    for (i = 0; i < 10; i++) {
        sum = sum + a[i];
    }
    printf("Sum of Array = %d\n", sum);
    return 0;
}
```

Output :

ain.c	  	Output
<pre>#include <stdio.h> int main() { int a[10], i; int sum = 0; printf("Enter the numbers in the array:\n"); for (i = 0; i < 10; i++) { scanf("%d", &a[i]); } for (i = 0; i < 10; i++) { sum = sum + a[i]; } printf("Sum of Array = %d\n", sum); return 0; }</pre>		<pre>/tmp/wwixho0U4.o Enter the numbers in the array: 1 6 7 8 9 1 2 6 7 9 Sum of Array = 56</pre>

3.

Ans.

```
#include <stdio.h>
int main() {
    int matrix1[3][3], matrix2[3][3], result[3][3];
    printf("Enter elements of the first matrix:\n");
    for (int i = 0; i < 3; i++) {
```

```

for (int j = 0; j < 3; j++) {
scanf("%d", &matrix1[i][j]);
}
}
printf("Enter elements of the second matrix:\n");
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
scanf("%d", &matrix2[i][j]);
}
}
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
result[i][j] = matrix1[i][j] + matrix2[i][j];
}
}
printf("Resultant matrix after addition:\n");
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
printf("%d ", result[i][j]);
}
printf("\n");
}
return 0;
}

```

Output :

ain.c	Run	Output
<pre> #include <stdio.h> int main() { int matrix1[3][3], matrix2[3][3], result[3][3]; printf("Enter elements of the first matrix:\n"); for (int i = 0; i < 3; i++) { for (int j = 0; j < 3; j++) { scanf("%d", &matrix1[i][j]); } } printf("Enter elements of the second matrix:\n"); for (int i = 0; i < 3; i++) { for (int j = 0; j < 3; j++) { scanf("%d", &matrix2[i][j]); } } for (int i = 0; i < 3; i++) { for (int j = 0; j < 3; j++) { result[i][j] = matrix1[i][j] + matrix2[i][j]; } } printf("Resultant matrix after addition:\n"); for (int i = 0; i < 3; i++) { for (int j = 0; j < 3; j++) { printf("%d ", result[i][j]); } printf("\n"); } return 0; } </pre>		<pre> /tmp/wwixho0U4.o Enter elements of the first matrix: 2 3 4 4 5 6 6 7 8 Enter elements of the second matrix: 1 2 8 2 4 6 9 0 4 Resultant matrix after addition: 3 5 12 6 9 12 15 7 12 </pre>

4.

Ans.

```
#include <stdio.h>
int main() {
int array[10];
int sum = 0;
int evenSum = 0;
int oddSum = 0;
printf("Enter 10 integers:\n");
for (int i = 0; i < 10; i++) {
scanf("%d", &array[i]);
sum += array[i];
if (array[i] % 2 == 0) {
evenSum += array[i];
} else {
oddSum += array[i];
}
}
printf("Sum of all elements: %d\n", sum);
printf("Sum of even elements: %d\n", evenSum);
printf("Sum of odd elements: %d\n", oddSum);
return 0;
}
```

Output :

in.c	Run	Output
<pre>#include <stdio.h> int main() { int array[10]; int sum = 0; int evenSum = 0; int oddSum = 0; printf("Enter 10 integers:\n"); for (int i = 0; i < 10; i++) { scanf("%d", &array[i]); sum += array[i]; if (array[i] % 2 == 0) { evenSum += array[i]; } else { oddSum += array[i]; } } printf("Sum of all elements: %d\n", sum); printf("Sum of even elements: %d\n", evenSum); printf("Sum of odd elements: %d\n", oddSum); return 0; }</pre>		<pre>/tmp/WWWixhoOU4.o Enter 10 integers: 3 4 5 6 76 8 5 3 2 1 3 Sum of all elements: 110 Sum of even elements: 90 Sum of odd elements: 20 </pre>

5.

Ans.

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

```
int main() {  
    printf("Size of char: %zu bytes\n", sizeof(char));  
    printf("Size of short: %zu bytes\n", sizeof(short));  
    printf("Size of int: %zu bytes\n", sizeof(int));  
    printf("Size of long: %zu bytes\n", sizeof(long));  
    printf("Size of long long: %zu bytes\n", sizeof(long long));  
    printf("Size of float: %zu bytes\n", sizeof(float));  
    printf("Size of double: %zu bytes\n", sizeof(double));  
    printf("Size of long double: %zu bytes\n", sizeof(long double));  
    printf("Size of unsigned char: %zu bytes\n", sizeof(unsigned char));  
    printf("Size of unsigned short: %zu bytes\n", sizeof(unsigned short));  
    printf("Size of unsigned int: %zu bytes\n", sizeof(unsigned int));  
    printf("Size of unsigned long: %zu bytes\n", sizeof(unsigned long));  
    printf("Size of unsigned long long: %zu bytes\n", sizeof(unsigned long long));  
    printf("Size of char*: %zu bytes\n", sizeof(char *));  
    printf("Size of int*: %zu bytes\n", sizeof(int *));  
    printf("Size of float*: %zu bytes\n", sizeof(float *));  
    printf("Size of double*: %zu bytes\n", sizeof(double *));  
    printf("Size of void*: %zu bytes\n", sizeof(void *));  
    printf("Size of enum: %zu bytes\n", sizeof(enum { RED, GREEN, BLUE }));  
    printf("Size of struct: %zu bytes\n", sizeof(struct { int x; double y; }));  
  
    return 0;  
}
```

Output.

main.c			Run	Output
<pre>#include <stdio.h> int main() { printf("Size of char: %zu bytes\n", sizeof(char)); printf("Size of short: %zu bytes\n", sizeof(short)); printf("Size of int: %zu bytes\n", sizeof(int)); printf("Size of long: %zu bytes\n", sizeof(long)); printf("Size of long long: %zu bytes\n", sizeof(long long)); printf("Size of float: %zu bytes\n", sizeof(float)); printf("Size of double: %zu bytes\n", sizeof(double)); printf("Size of long double: %zu bytes\n", sizeof(long double)); printf("Size of unsigned char: %zu bytes\n", sizeof(unsigned char)); printf("Size of unsigned short: %zu bytes\n", sizeof(unsigned short)); printf("Size of unsigned int: %zu bytes\n", sizeof(unsigned int)); printf("Size of unsigned long: %zu bytes\n", sizeof(unsigned long)); printf("Size of unsigned long long: %zu bytes\n", sizeof(unsigned long long)); printf("Size of char*: %zu bytes\n", sizeof(char *)); printf("Size of int*: %zu bytes\n", sizeof(int *)); printf("Size of float*: %zu bytes\n", sizeof(float *)); printf("Size of double*: %zu bytes\n", sizeof(double *)); printf("Size of void*: %zu bytes\n", sizeof(void *)); printf("Size of enum: %zu bytes\n", sizeof(enum { RED, GREEN, BLUE })); printf("Size of struct: %zu bytes\n", sizeof(struct { int x; double y; })); return 0; }</pre>	<pre>/tmp/z3vB7HDqTY.o Size of char: 1 bytes Size of short: 2 bytes Size of int: 4 bytes Size of long: 8 bytes Size of long long: 8 bytes Size of float: 4 bytes Size of double: 8 bytes Size of long double: 16 bytes Size of unsigned char: 1 bytes Size of unsigned short: 2 bytes Size of unsigned int: 4 bytes Size of unsigned long: 8 bytes Size of unsigned long long: 8 bytes Size of char*: 8 bytes Size of int*: 8 bytes Size of float*: 8 bytes Size of double*: 8 bytes Size of void*: 8 bytes Size of enum: 4 bytes Size of struct: 16 bytes</pre>			