INDEX

Sr. No	Name of the Experiment	Page No	Date of Experiment	Date of Submission	Remarks
1	Experiment-6 Function 5th question(WAP)	01		25/11/25	
2	Experiment-7 Structure and Union	02		25/11/25	
3	Experiment-8 Pointers WAP questions	08		25/11/25	
4	Experiment-9 File handling in C and WAP question ->	11		25/11/25	
5)	Experiment-10 Dynamic memory allocation	14		25/11/25	
6)	Experiment-11 Bitwise operator	18		25/11/25	
7)	Experiment-12 Preprocess and directive is C	20		25/11/25	

INDEX

Sr. No	Name of the Experiment	Page No	Date of Experiment	Date of Submission	Remarks
8)	Experiment – 13 Macros in C	21		25/11/25	
9)	Experiment – 14 State library in C.	22		25/11/25	
10)					

Date

Expt. No. _____ Page No. 1

5) Input code:-

```
#include <stdio.h>
#include <string.h>
void REVERSE (char str[]) {
    int i, j;
    char temp;
    int len = strlen (str);
    for (i=0, j= len-1; i<j; i++, j--) {
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
    }
}

int main() {
    char str[100];
    printf ("Enter a string: ");
    scanf ("%s", str);

    REVERSE(str);

    printf("Reversed string: %s\n", str);

    return 0;
}
```

Teacher's Signature : _____

Output :-

Enter a string : hello
Reversed string : olleh

Date

Expt. No. _____ Page No. 2

Experiment :7: Structures and Union

1) Input Code:-

```
#include <stdio.h>

struct complex {
    float real, imag;
};

int main() {
    struct complex a, b, sum, difference;

    printf("Enter real and imaginary parts of first number:");
    scanf("%f %f", &a.real, &a.imag);

    printf("Enter real and imaginary parts of second number:");
    scanf("%f %f", &b.real, &b.imag);

    sum.real = a.real + b.real;
    sum.imag = a.img + b.imag;

    diff.real = a.real - b.real;
    diff.real = a.imag - b.imag;
```

Date

Expt. No.

Page No. 3

```
printf("Sum : %0.2f %c %0.2fi\n", sum.real, (sum.imag > 0 ? '+' : '-'), (sum.imag > 0 ? sum.imag : -sum.imag));

printf("Difference : %0.2f %c %0.2fi\n", diff.real, (diff.imag > 0 ? '+' : '-'), (diff.imag > 0 ? diff.imag : -diff.imag));

return 0;
}
```

Output :-

Enter real and imaginary parts of first number : 3 2
Enter real and imaginary parts of second number : 1 4
Sum 4.00 + 6.00i
Difference : 2.00 - 2.00i

2) Input Code:-

```
#include <stdio.h>

struct Employee {
    char name[50];
    float basic, da, gross;
};

int main() {
struct Employee emp[100];
int n, i;

printf("Enter number of employes: ");
scanf("%d", &n);

for(i=0; i<n; i++) {
  printf("\nEnter name of employes %d: "; i+1);
  scanf("%s", emp[i].name);
  printf("Enter basic pay: ");
  scanf("%f"; &emp[i].basic);
  emp[i]da = 0.52 * emp[i].basic;
  emp[i].gross = emp[i].basic + emp[i].da;
}

printf("\n Employee Name \t Gross Salary \n");
```

Date

Expt. No. Page No. 5

```
for(i=0; i<n; i++ ){
    printf("%s\t%.2f\n", emp[i].name, emp[i].gross);
}
    return 0;
}
```

3) Input Code:-

```
#include <stdio.h>

struct Book {
    int book_id;
    char title[50];
    char author[50];
    float price;
};
void displayBook(struct Book b) {
    printf("\n Book ID: %d\n", b.book_id);
    printf("Title : %s\n", b.title);
    printf("Author : %s \n", b.author);
    printf("Price: %.2f\n", b.price);
}
int main() {
    struct Book b1;
printf("Enter Book ID :");
scanf("%d", &b1.book_id);
printf("Enter Title ;");
scanf("%[^\n]", b1.title);
printf("Enter Author :");
scanf("%[^\n]", b1.author);
printf("Enter Price : ");
scanf("%f", &b1.price);
```

Teacher's Signature :

```
    displayBook(b1);
    return 0;
}
```

Expt. No. _____ Date _____

Page No. 8

Experiment 8: Pointers

1) Input Code:-

```
#include <stdio.h>
int main() {
    int a = 10;
    float b = 3.5;
    char c = 'x';

    int *p1 = &a;
    float *p2 = &b;
    char *p3 = &c;

    printf("Value of a: %d, via pointer: %d\n",
           a, *p1);
    printf("Value of b: %.2f, via pointer =
           %.2f\n", b, *p2);
    printf("Value of c: %c via pointer: %c\n", c, *p3);

    return 0;
}
```

Date

Expt. No. _____ Page No. 9

2) Input Code:-

```
#include <stdio.h>
int main() {
    int a = 5;
    int *p = &a;

    printf("Address = %p\n", p);
    p++;
    printf("After increament = %p\n", p);
    p--;
    printf("After decrement = %p\n", p);
    return 0;

}
```

Expt. No. _____ Date _____

Page No. 10

3) ⊙ Input Code:-

```
#include <stdio.h>

void update (int *x, int *y) {
    *x = *x + 5;
    *y = *y + 10;
}

int main () {
    int a = 10, b = 20;
    update (&a, &b);
    printf ("Updated values : a = %d, b = %d \n", a, b);

    return 0;
}
```

Date

Expt. No.

Page No. 11

Experment 9 : file handling in C

1) ⊙ Input Code :-

```
#include <stdio.h>
int main() {
FILE *fp = fopen("sample.txt", "w");
fprintf(fp, "This is a test file.");
fclose(fp);
return 0;
}
```

Date

Expt. No.

Page No. 12

2) Input Code:-

```
#include <stdio.h>
int main()
{
FILE *fp = fopen("sample.txt", "r");
char ch;
while ((ch = fgetc(fp)) != EOF)
printf("%c", ch);
fclose(fp);
return 0;
}
```

Expt. No. _____ Date _____

Page No. 13

3/2 Input Code

```
#include <stdio.h>
int main () {
FILE * fp = fopen ("sample.txt", "r");
char line [200];
while (fgets (line, sizeof (line), fp))
printf ("%s", line);
fclose (fp);
return 0;
}
```

Date

Expt. No. _____ Page No. 124

Experiment 10:- Dynamic memory allocation

1) ~~#include <stdio~~

Input code:-

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

Struct Node{
  int data;
  struct Node* next;
};
int main(){
struct node *head = NULL; *newNode, *temp;
int n, value;
printf("Enter number of nodes:");
Scanf("%d", &n);

for(int i=0; i<n; i++){
    Printf("Enter value:");
    Scanf("%d", &value);
  Newnode = (struct Node *) malloc(
      sizeof(struct node));
  New node -> data = value;
  new node -> next = NULL;
```

Teacher's Signature : _____

Date

Expt. No. _____ Page No. 15

```
if (head == NULL) head = newNode;
else {
    temp = head;
    while (temp->next != NULL) temp = temp->next;
    temp->next = newnode;
}
}

printf("Linked list :");
temp = head;
while (temp != NULL) {
    printf("%d ->", temp->data);
    temp = temp->next;
}
return 0;
}
```

Date

Expt. No. _____ Page No. 16

2) WAP to insert item in middle of the Linked List

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

struct Node {
    int data;
    struct Node * next;
};

int main () {
    struct Node * head = NULL; * new Node, * temp;
    int pos, value;

    for (int i = 1; i <= 3; i++) {
        new Node = (struct Node *) malloc (sizeof (struct Node
        ));
        new Node -> data = i * 10;
        new Node -> next = next;
        head = new Node;
    }

    printf ("Enter value to insert : ");
    scanf ("%d", &value);
    printf ("Enter position : ");
    scanf ("%d", &pos);
```

Teacher's Signature : _____

Date

Expt. No. _____ Page No. 17

```
newNode = (struct Node *) malloc (sizeof(struct Node));
newNode -> data = value;

temp = head;
for (int i = 1; i < pos - 1; i++) temp = temp -> next;

newNode -> next = temp -> next;
temp -> next = newNodes;

printf ("Updated List :");
temp = head;
while (temp -> 5 = NULL) {
  printf ("%d -> ", temp -> data);
  temp = temp -> next;
}
  return 0;
}
```

Expt. No. _____ Date _____

Page No. 18

Experiment 11: Bitwise Operator

1) WAP to apply bitwise OR, AND and NOT operators on bit level.

→

```
#include <stdio.h>
int main ( ) {
  int a = 5, b = 3;
  printf ("a & b = %d \n", a & b);
  printf ("a | b = %d \n", a | b);
  printf ("~a = %d \n" a ~a);
  printf (" a<<1 = %d \n", a << 1);
  printf (" a>>1 = %d \n", a >> 1);

  return 0;

}
```

Expt. No. _____ Date _____

Page No. 19

2) WAP to apply left shift and right shift

→

```
#include <stdio.h>

int main()
{
int num, ls, rs;

printf("Enter a number : ");
scanf("%d", &num);

ls = num << 1;
rs = num >> 1;

printf("Original number = %d \n", num);
printf("After left shift (num<<1) = %d \n", ls);
Printf("After right shift (num>>1) = %d \n", rs);

return 0;

}
```

Expt. No. _____ Date _____

Page No. 20

Experiment 12: Preprocessor and Directives in C

1) WAP to define some constant variable in preprocessor.

```
#include <stdio.h>
#define PI 3.1415g

int main() {
    float r, area;
    printf("Enter radius: ");
    scanf("%f", &r);

    area = PI * r * r;

    printf("Area of circle = %.2f\n", area);

    return 0;
}
```

Expt. No. _____ Date _____

Page No. 21

2) WAP of Define a function in different ways.

→

```
#include <stdio.h>
#define Square(x) ((x) * (x))

int main ()
{
    int num;

    printf ("Enter a number:");
    scanf ("%d", &num);

    printf ("Square of %d is %d\n", num, Square
                                        (num));

    return 0;

}
```

Expt. No. _____ Date _____

Page No. 21

Experiment 13 Macros in C.

11/2 WAP to define multiple macro to perform arithmetic functions.

→
```
#include <stdio.h>
#define ADD (a,b) ((a)+(b))
#define SUB (a,b) ((a)-(b))
#define MUL (a,b) ((a)*(b))
#define DIV (a,b) ((b)!=0 ?(a)/(b):0)

int main ()
{
  int x,y;
  printf ("Enter two numbers : ");
  Scanf ("%d %d", &x, &y);
  printf ("Addition : %d \n" ADD (x,y));
  printf ("Substraction : %d\n, SUB (x,y));
  printf ("Multiplication : %d\n", MUL (x,y));
  printf ("Division : %d \n", DIV (x,y));

  return 0;
}
```

Experiment 14:- Static library in C

1) WAP to create static library to perform arithmetic functions.

→

```
#ifndef ARITH_H
#define ARITH_H
int add(int, int);
int sub(int, int);
int mul(int, int);
float divi(int, int);
#endif
```

```
#include "arith.h"
int add(int a, int b) {return a+b;}
int sub(int a, int b) {return a-b;}
int mul(int a, int b) {return a*b;}
float divi(int a, int b)
{return b? (float)a/b: 0;}
```

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

int main()
{
    int x=10, y=5;
    printf("Add: %d\n", add(x,y));
    printf("Sub: %d\n", sub(x,y));
    printf("Mul: %d\n", mul(x,y));
    printf("Div: %0.2f\n", divi(x,y));
    return 0;
}
```

Teacher's Signature :

Expt. No. _____ Date _____

Page No. 23

2) WAP to use static library in other program.

```
#ifndef ARITH_H
#define ARITH_H

int add(int, int);
int sub(int, int);
int mul(int, int);
float divi(int, int);
```

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

int main() {
    int a = 20, b = 4;
    printf("Add : %d \n", add(a,b));
    printf("Sub : %d \n", sub(a,b));
    printf("Mul : %d \n", mul(a,b));
    printf("Div : %.2f \n", divi(a,b));

    return 0;
}
```