C PROGRAM LAB

**Set 1**

1. **Triangle Area:**

#include <stdio.h>

int main() {

float base, height, area;

printf("Enter base and height: ");

scanf("%f %f", &base, &height);

area = 0.5 \* base \* height;

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

return 0;

}

1. **No Square Area using Macro:**

#include <stdio.h>

#define AREA(x) (x \* x)

int main() {

int side;

printf("Enter side of square: ");

scanf("%d", &side);

printf("Area = %d\n", AREA(side));

return 0;

}

**Set 2**

1. **Celsius to Fahrenheit:**

#include <stdio.h>

int main() {

float c, f;

printf("Enter Celsius: ");

scanf("%f", &c);

f = (c \* 9/5) + 32;

printf("Fahrenheit = %.2f\n", f);

return 0;

}

1. **File Copy Program:**

#include <stdio.h>

int main() {

FILE \*f1 = fopen("SampleTextFile1.txt", "w");

char ch;

printf("Enter text ('@' to stop):\n");

while ((ch = getchar()) != '@') fputc(ch, f1);

fclose(f1);

f1 = fopen("SampleTextFile1.txt", "r");

FILE \*f2 = fopen("SampleTextFile2.txt", "w");

while ((ch = fgetc(f1)) != EOF) fputc(ch, f2);

fclose(f1); fclose(f2);

f2 = fopen("SampleTextFile2.txt", "r");

printf("\nContents of copied file:\n");

while ((ch = fgetc(f2)) != EOF) putchar(ch);

fclose(f2);

return 0;

}

**Set 3**

1. **Largest of Three Numbers:**

#include <stdio.h>

int main() {

int a, b, c;

printf("Enter 3 numbers: ");

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

if (a >= b && a >= c)

printf("Largest = %d\n", a);

else if (b >= a && b >= c)

printf("Largest = %d\n", b);

else

printf("Largest = %d\n", c);

return 0;

}

1. **Student Structure:**

#include <stdio.h>

struct Student {

char name[20];

int roll;

float marks;

};

int main() {

struct Student s[5];

for (int i = 0; i < 5; i++) {

printf("Enter name, roll, marks: ");

scanf("%s %d %f", s[i].name, &s[i].roll, &s[i].marks);

}

for (int i = 0; i < 5; i++)

printf("%s %d %.2f\n", s[i].name, s[i].roll, s[i].marks);

return 0;

}

**Set 4**

1. **Weekday using Switch-Case:**

#include <stdio.h>

int main() {

int day;

printf("Enter weekday number (1-7): ");

scanf("%d", &day);

switch(day) {

case 1: printf("Monday\n"); break;

case 2: printf("Tuesday\n"); break;

case 3: printf("Wednesday\n"); break;

case 4: printf("Thursday\n"); break;

case 5: printf("Friday\n"); break;

case 6: printf("Saturday\n"); break;

case 7: printf("Sunday\n"); break;

default: printf("Invalid\n");

}

return 0;

}

1. **Highest Paid Employee:**

#include <stdio.h>

struct Employee {

int id;

char name[20];

float salary;

};

int main() {

struct Employee e[3];

int i, max = 0;

for (i = 0; i < 3; i++) {

scanf("%d %s %f", &e[i].id, e[i].name, &e[i].salary);

if (e[i].salary > e[max].salary) max = i;

}

printf("%d\n%s\n%.2f\n", e[max].id, e[max].name, e[max].salary);

return 0;

}

**Set 5**

1. **Armstrong Number:**

#include <stdio.h>

int main() {

int num, sum = 0, temp, rem;

printf("Enter number: ");

scanf("%d", &num);

temp = num;

while (temp != 0) {

rem = temp % 10;

sum += rem \* rem \* rem;

temp /= 10;

}

if (sum == num)

printf("Armstrong Number\n");

else

printf("Not Armstrong\n");

return 0;

}

1. **Linear Search with Pointers:**

#include <stdio.h>

int main() {

int a[5], i, key, \*p;

printf("Enter 5 elements: ");

for (i = 0; i < 5; i++) scanf("%d", &a[i]);

printf("Enter element to search: ");

scanf("%d", &key);

p = a;

for (i = 0; i < 5; i++) {

if (\*(p + i) == key) {

printf("Element found at index %d\n", i);

return 0;

}

}

printf("Element not found\n");

return 0;

}

**Set 6**

1. **Sum of Squares:**

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Enter N: ");

scanf("%d", &n);

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

sum += i \* i;

printf("Sum = %d\n", sum);

return 0;

}

1. **Swap Using Pointers:**

#include <stdio.h>

void swap(int \*a, int \*b) {

int t = \*a;

\*a = \*b;

\*b = t;

}

int main() {

int x, y;

scanf("%d %d", &x, &y);

swap(&x, &y);

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

return 0;

}

**Set 7:**

1. **Check Prime**

#include <stdio.h>

int main() {

int n, isPrime = 1;

scanf("%d", &n);

if(n <= 1) isPrime = 0;

for(int i=2; i<=n/2; i++)

if(n % i == 0) { isPrime = 0; break; }

printf("%s", isPrime ? "Prime" : "Not Prime");

return 0;

}

1. **String Functions**

#include <stdio.h>

#include <string.h>

int main() {

char str1[100], str2[100];

scanf("%s %s", str1, str2);

printf("Length: %lu and %lu\n", strlen(str1), strlen(str2));

printf("Compare: %d\n", strcmp(str1, str2));

strcat(str1, str2);

printf("Concatenated: %s\n", str1);

return 0;

}

**Set 8:**

1. **Multiplication Tables up to n**

#include <stdio.h>

int main() {

int n;

scanf("%d", &n);

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

for(int j = 1; j <= 10; j++) {

printf("%d x %d = %d\n", i, j, i\*j);

}

}

return 0;

}

1. **Check Palindrome (case-insensitive)**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

int main() {

char str[100], clean[100];

int j = 0;

scanf("%s", str);

for(int i = 0; str[i]; i++)

if(isalpha(str[i])) clean[j++] = tolower(str[i]);

clean[j] = '\0';

int isPalin = 1;

for(int i = 0; i < j/2; i++)

if(clean[i] != clean[j-1-i]) { isPalin = 0; break; }

printf("%s", isPalin ? "Palindrome" : "Not Palindrome");

return 0;

}

**Set 09:**

1. **Decimal to Octal**

#include <stdio.h>

int main() {

int n;

scanf("%d", &n);

printf("%o", n);

return 0;

}

1. **Recursive Factorial**

#include <stdio.h>

int factorial(int n) {

if(n <= 1) return 1;

return n \* factorial(n - 1);

}

int main() {

int n;

scanf("%d", &n);

printf("%d", factorial(n));

return 0;

}

**Set 10**:

1. **Read Integer Array & Compute Sum**

#include <stdio.h>

int main() {

int n, a[100], sum = 0;

scanf("%d", &n);

for(int i = 0; i < n; i++) {

scanf("%d", &a[i]);

sum += a[i];

}

printf("Sum = %d", sum);

return 0;

}

1. **Matrix Multiplication with Validity Check**

#include <stdio.h>

int main() {

int r1, c1, r2, c2;

scanf("%d %d", &r1, &c1);

int A[r1][c1];

for(int i = 0; i < r1; i++)

for(int j = 0; j < c1; j++)

scanf("%d", &A[i][j]);

scanf("%d %d", &r2, &c2);

int B[r2][c2];

for(int i = 0; i < r2; i++)

for(int j = 0; j < c2; j++)

scanf("%d", &B[i][j]);

if(c1 != r2) {

printf("Not Possible");

return 0;

}

int result[r1][c2];

for(int i = 0; i < r1; i++)

for(int j = 0; j < c2; j++) {

result[i][j] = 0;

for(int k = 0; k < c1; k++)

result[i][j] += A[i][k] \* B[k][j];

}

for(int i = 0; i < r1; i++) {

for(int j = 0; j < c2; j++)

printf("%d ", result[i][j]);

printf("\n");

}

return 0;

}

**Set 11:**

1. **Average of Five Integers (2 Decimal Places)**

#include <stdio.h>

int main() {

int a[5], sum = 0;

for(int i = 0; i < 5; i++) {

scanf("%d", &a[i]);

sum += a[i];

}

float avg = sum / 5.0;

printf("%.2f", avg);

return 0;

}

1. **Linear Search (Without Pointers)**

#include <stdio.h>

int main() {

int n, key, a[100];

scanf("%d", &n);

for(int i = 0; i < n; i++)

scanf("%d", &a[i]);

scanf("%d", &key);

for(int i = 0; i < n; i++) {

if(a[i] == key) {

printf("Found");

return 0;

}

}

printf("Not Fou

nd");

return 0;

}