

**PSG COLLEGE OF TECHNOLOGY, COIMBATORE -641004 DEPARTMENT OF
COMPUTER APPLICATIONS I SEMESTER MCA
23MX16 C PROGRAMMING LABORATORY
PROBLEM SHEET 1 - WEEK 1**

Tino Britty J (D25MX316)

1. Write a C program to compute the following

```
#include<stdio.h>
#include<conio.h>
int main(){
float x,f;
printf("Enter the x value : ");
scanf("%f",&x);
f=(x*x*x-2*x*x+x-6.3)/(x*x+0.05*x+3.14);
printf("%f",f);
return 0;
}
```

2. Write a C program to print the roots of a quadratic equation

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

int main() {
    double a, b, c, discriminant, root1, root2, realPart, imagPart;

    printf("Enter coefficients a, b and c: ");
    scanf("%lf %lf %lf", &a, &b, &c);

    discriminant = b * b - 4 * a * c;
```

```

if (discriminant > 0) {
    root1 = (-b + sqrt(discriminant)) / (2 * a);
    root2 = (-b - sqrt(discriminant)) / (2 * a);
    printf("Roots are real and different.\n");
    printf("Root1 = %.2lf , Root2 = %.2lf\n", root1, root2);
}

else if (discriminant == 0) {
    root1 = root2 = -b / (2 * a);
    printf("Roots are real and equal.\n");
    printf("Root1 = Root2 = %.2lf\n", root1);
}

else {
    realPart = -b / (2 * a);
    imagPart = sqrt(-discriminant) / (2 * a);
    printf("Roots are complex and different.\n");
    printf("Root1 = %.2lf + %.2lfi , Root2 = %.2lf - %.2lfi\n",
           realPart, imagPart, realPart, imagPart);
}

return 0;
}

```

- 3. Write the complete C program that asks the user to enter the value for t from the keyboard and then it computes and prints the value of p which is expressed as a function of t by**

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

```

int main()
{
    float t,p;
    printf("Enter the value: ");
    scanf("%f",&t);
    if(t>0&&t<=2)
        p=20;
    else if((t>13&&t<=16) || t>30)
        p= 4*(t+2);
    else
        p=4*(t*t+2*t);
    printf("%f",p);
    return 0;
}

```

4. Write a program that outputs the day of the week given a date expressed as j (day) m (month) a (year). You will use the following formula:

```

#include <stdio.h>

int main() {
    int day, month, year, century, yearInCentury;
    int result, remainder;
    char *days[] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
    "Friday", "Saturday"};

    printf("Enter date (DD MM YYYY): ");
    scanf("%d %d %d", &day, &month, &year);

    century = year / 100;

```

```

yearInCentury = year % 100;

result = (century / 4) - (2 * century) - 1;
result += (5 * yearInCentury) / 4;
result += (26 * (month + 1)) / 10;
result += day;

remainder = result % 7;
if (remainder < 0) {
    remainder += 7;
}

printf("The day is: %s\n", days[remainder]);

return 0;
}

```

5. Write a program to find the biggest among two numbers without using any control structures.

```

#include<stdio.h>
#include<conio.h>

void main()
{
    int a,b,bigest;
    printf("Enter the value of a,b: ");
    scanf("%d%d",&a,&b);
    bigest=((a+b)+abs(a-b))/2;
    printf("%d",bigest);
}

```

```
    return 0;
```

```
}
```

6. Write a C program to print * in the format given below.

```
#include <stdio.h>

int main() {
    int n = 5;
    for(int i = 1; i <= n; i++) {
        for(int j = 1; j <= n - i; j++) {
            printf(" ");
        }
        for(int k = 1; k <= 2 * i - 1; k++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

7. Write a program for a matchstick game being played between the computer and a user. Your program should ensure that the computer always wins. Rules for the game are as follows:

- There are 21 matchsticks.
- The computer asks the player to pick 1, 2, 3, or 4 matchsticks.
- After the person picks, the computer does its picking. - Whoever is forced to pick up the last matchstick loses the game.

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

```
int main() {
```

```
    int total = 21;
```

```
int user_pick, computer_pick;

printf("There are 21 matchsticks.\n");
printf("You can pick 1, 2, 3, or 4 matchsticks each turn.\n");

while (total > 1) {
    do {
        printf("Your turn. Pick 1 to 4 matchsticks: ");
        scanf("%d", &user_pick);
        if (user_pick < 1 || user_pick > 4 || user_pick > total - 1) {
            printf("Invalid pick. Try again.\n");
        }
    } while (user_pick < 1 || user_pick > 4 || user_pick > total - 1);

    total -= user_pick;
    printf("Matchsticks left: %d\n", total);

    if (total == 1) {
        printf("You lose! Computer wins \n");
        break;
    }

    computer_pick = 5 - user_pick;

    if (computer_pick < 1 || computer_pick > 4) {
        computer_pick = 1;
    }
}
```

```

total -= computer_pick;
printf("Computer picks %d matchstick(s).\n", computer_pick);
printf("Matchsticks left: %d\n", total);

if (total == 1) {
    printf("You are forced to pick the last matchstick. You lose!\n");
    break;
}

return 0;
}

```

8. A number is said to be perfect if it is equal to the sum of all numbers which are its

factors (excluding itself). So, for example, 6 is perfect, because it is the sum of its factors 1,2,3. Write a program which determines if a number is perfect. It should also print its factors.

```

#include <stdio.h>

int main() {
    int num, i, sum = 0;

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

    printf("Factors of %d: ", num);

```

```

for (i = 1; i <= num / 2; i++) {
    if (num % i == 0) {
        printf("%d ", i);
        sum += i;
    }
}
printf("\n");

if (sum == num) {
    printf("%d is a perfect number.\n", num);
} else {
    printf("%d is NOT a perfect number.\n", num);
}

return 0;
}

```

9. Write a program that takes as input a natural number x and prints the smallest palindrome larger than x.

```

#include <stdio.h>

#include <stdbool.h>

bool isPalindrome(int num) {
    int reversed = 0, original = num;

    while (num > 0) {
        reversed = reversed * 10 + (num % 10);
        num /= 10;
    }
}
```

```

    }

    return (original == reversed);
}

int main() {
    int x;

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

    int mit = x - 1;

    while (!isPalindrome(mit)) {
        mit--;
    }

    printf("The next smallest palindrome after %d is %d.\n", x, mit);

    return 0;
}

```

10. According to a study, the approximate level of intelligence of a person can be calculated

using the following formula: $i = 2 + (y + 0.5x)$

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

```
int main() {
```

```
    double x, i;
```

```
int y;  
printf(" y  x  i\n");  
printf("-----\n");  
for (y = 1; y <= 6; y++) {  
    for (x = 5.5; x <= 12.5; x += 0.5) {  
        i = 2 + (y + 0.5 * x);  
        printf("%3d %5.1f %6.2f\n", y, x, i);  
    }  
}  
  
return 0;  
}
```