

NAME: SOHAM BANIK
DEPARTMENT: ELECTRONICS AND TELECOMMUNICATION ENGINEERING
ROLL NUMBER: 002310701006
SESSION: 2023-2024, ODD SEMESTER
SUBJECT: COMPUTER PROGRAMMING & NUMERICAL METHODS

ASSIGNMENT 3

QUESTION 1:

Write a menu driven C program that prints the following pattern for input 'n' by the user.

SOLUTION:

```
#include <stdio.h>
int main()
{
    int n;
    printf("Enter size of pattern-");
    scanf("%d", &n);
    printf("\n");
    // Pattern 1
    for (int i = 1; i < n + 1; i++)
    {
        for (int j = 1; j < i + 1; j++)
        {
            printf("%d ", j);
        }
        printf("\n");
    }
    printf("\n");
    // Pattern 2
    for (int i = 1; i < n + 1; i++)
    {
        for (int j = 1; j < n - i + 1; j++)
        {
            printf(" ");
        }
        for (int j = 1; j < i + 1; j++)
        {
            printf("%d", j);
        }
        for (int j = i; j > 0 + 1; j--)
        {
            printf("%d", j - 1);
        }
        printf("\n");
    }
}
```

```

}
printf("\n");
// Pattern 3
for (int i = n; i > 1; i--)
{
    for (int j = i; j > n - i - 1; j--)
    {
        printf("%d ", j);
    }
    printf("\n");
}
printf("\n");
// Pattern 4
for (int i = 1; i < 2 * n; i++)
{
    for (int j = 1; j < 2 * n; j++)
    {
        if (i + j == n + 1 || i - j == n - 1 || j - i == n - 1 || i + j == 3 * n - 1)
        {
            printf("+");
        }
        else
        {
            printf(" ");
        }
    }
    printf("\n");
}
return 0;
}

```

OUTPUT:

```

PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 1.c -o 1 } ; if ($?) { .\1 }
Enter size of pattern-4

1
1 2
1 2 3
1 2 3 4

    1
   121
  12321
 1234321

4 3 2 1 0
3 2 1
2

+
++
+ +
+  +
+  +
++
+

PS F:\C3>

```

QUESTION 2:

Write a menu driven C program with recursive function to print factorial of a number given by the user. The program must repeatedly ask for numbers unless the user inputs -1.

SOLUTION:

```
#include <stdio.h>
int fact(int n);
int main()
{
    int n;
    while (1)
    {
        printf("\nEnter a number: ");
        scanf("%d", &n);
        if (n < -1)
        {
            printf("Enter a positive integer!");
        }
        else if (n == -1)
        {
            break;
        }
        else
        {
            printf("Factorial of %d is %d", n, fact(n));
        }
    }
    return 0;
}
int fact(int n)
{
    if (n == 0)
    {
        return 1;
    }
    else
    {
        return n * fact(n - 1);
    }
}
```

OUTPUT:

```
PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 2.c -o 2 } ; if ($?) { .\2 }

Enter a number: 3
Factorial of 3 is 6
Enter a number: 5
Factorial of 5 is 120
Enter a number: -8
Enter a positive integer!
Enter a number: -1
PS F:\C3>
```

QUESTION 3:

Write a C program with recursive function to print Fibonacci series of first N numbers where N is the user input.

SOLUTION:

```
#include <stdio.h>
int fibo(int n)
{
    if (n <= 1)
        return n;
    return fibo(n - 1) + fibo(n - 2);
}
void printer(int n)
{
    if (n <= 0)
        return;

    for (int i = 0; i < n; i++)
    {
        printf("%d ", fibo(i));
    }
}

int main()
{
    int n;

    printf("Enter the number of terms for Fibonacci series: ");
    scanf("%d", &n);

    printf("Fibonacci Series up to %d terms: \n", n);
    printer(n);

    return 0;
}
```

OUTPUT:

```
PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 3.c -o 3 } ; if ($?) { .\3 }
Enter the number of terms for Fibonacci series: 8
Fibonacci Series up to 8 terms:
0 1 1 2 3 5 8 13
PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 3.c -o 3 } ; if ($?) { .\3 }
Enter the number of terms for Fibonacci series: 10
Fibonacci Series up to 10 terms:
0 1 1 2 3 5 8 13 21 34
PS F:\C3> 
```

QUESTION 4:

Write a menu driven C program to check whether a positive integer not containing the number zero is palindrome or not. Use proper error checking. Do not use arrays.

SOLUTION:

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

int main()
{
    int n;
    printf("Enter a number without using '0': ");
    scanf("%d", &n);

    int flag = 0;
    int temp = n;
    while (temp > 0)
    {
        int digit = temp % 10;
        if (digit == 0)
        {
            flag = 1;
            break;
        }
        temp /= 10;
    }

    if (flag == 1)
    {
        printf("Your Number has a zero in it!\n");
    }
    else
    {
        int reverse = 0;
        temp = n;
        while (temp > 0)
        {
            int digit = temp % 10;
            reverse = reverse * 10 + digit;
            temp /= 10;
        }
        if (reverse != n)
        {
            printf("%d is not a palindrome!\n", n);
        }
        else
        {
            printf("%d is a palindrome!\n", n);
        }
    }
}
```

```
    return 0;
}
```

OUTPUT:

```
PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 4.c -o 4 } ; if ($?) { .\4 }
Enter a number without using '0': 12321
12321 is a palindrome!
PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 4.c -o 4 } ; if ($?) { .\4 }
Enter a number without using '0': 8558
8558 is a palindrome!
PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 4.c -o 4 } ; if ($?) { .\4 }
Enter a number without using '0': 42069
Your Number has a zero in it!
PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 4.c -o 4 } ; if ($?) { .\4 }
Enter a number without using '0': 4556
4556 is not a palindrome!
PS F:\C3> █
```

QUESTION 5:

Write a C program which accepts a number 'n' and prints all prime factors of n.

SOLUTION:

```
#include <stdio.h>
#include <math.h>
int main()
{
    int n;
    printf("Enter a positive number : ");
    scanf("%d", &n);
    printf("The prime factors of %d are:\n", n);
    for (int i = 2; i <= n / 2; i++)
    {
        if (n % i == 0)
        {
            int flag = 0;
            for (int j = 2; j <= sqrt(i); j++)
            {
                if (i % j == 0)
                {
                    flag = 1;
                    break;
                }
            }
            if (flag == 0)
            {
                printf("%d\n", i);
            }
        }
    }
}
```

```

    }
}

return 0;
}

```

OUTPUT:

```

PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 5.c -o 5 } ; if ($?) { .\5 }
Enter a positive number : 120
The prime factors of 120 are:
2
3
5
PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 5.c -o 5 } ; if ($?) { .\5 }
Enter a positive number : 10101
The prime factors of 10101 are:
3
7
13
37
PS F:\C3> █

```

QUESTION 6:

Write a menu driven C program which takes a number in any system [BINARY, DECIMAL, OCTAL, HEXADECIMAL] , converts it into all other systems and displays it.

SOLUTION:

```

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>

int convertBinaryToDecimal(long long n) {
    int decimalNumber = 0, i = 0, remainder;
    while (n!=0) {
        remainder = n%10;
        n /= 10;
        decimalNumber += remainder*pow(2,i);
        ++i;
    }
    return decimalNumber;
}

int convertOctalToDecimal(int octalNumber) {
    int decimalNumber = 0, i = 0;
    while(octalNumber != 0) {
        decimalNumber += (octalNumber%10) * pow(8,i);
        ++i;
    }
}

```

```

        octalNumber/=10;
    }
    return decimalNumber;
}

```

```

int convertHexadecimalToDecimal(char hexVal[]) {
    int len = strlen(hexVal);
    int base = 1;
    int decimalNumber = 0;
    for (int i=len-1; i>=0; i--) {
        if (hexVal[i]>='0' && hexVal[i]<='9') {
            decimalNumber += (hexVal[i] - 48)*base;
            base = base * 16;
        }
        else if (hexVal[i]>='A' && hexVal[i]<='F') {
            decimalNumber += (hexVal[i] - 55)*base;
            base = base*16;
        }
    }
    return decimalNumber;
}

```

```

void decToBinary(int n) {
    int binaryNum[32];
    int i = 0;
    while (n > 0) {
        binaryNum[i] = n % 2;
        n = n / 2;
        i++;
    }
    printf("\nBinary: ");
    for (int j = i - 1; j >= 0; j--)
        printf("%d", binaryNum[j]);
}

```

```

void decToOctal(int n) {
    int octalNum[100];
    int i = 0;
    while (n != 0) {
        octalNum[i] = n % 8;
        n = n / 8;
        i++;
    }
    printf("\nOctal: ");
    for (int j = i - 1; j >= 0; j--)
        printf("%d", octalNum[j]);
}

```

```

void decToHexa(int n) {
    char hexaDeciNum[100];
    int i = 0;
    while(n!=0) {

```



```

    int temp = 0;
    temp = n % 16;
    if(temp < 10) {
        hexaDeciNum[i] = temp + 48;
        i++;
    }
    else {
        hexaDeciNum[i] = temp + 55;
        i++;
    }
    n = n/16;
}
printf("\nHexadecimal: ");
for(int j=i-1; j>=0; j--)
    printf("%c", hexaDeciNum[j]);
}

int main() {
    int choice, num;
    long long binaryNum;
    int octalNum;
    char hexNum[17];
    printf("Enter the number: ");
    scanf("%d", &num);
    printf("Choose the number system of the input number: \n1. Binary\n2. Decimal\n3. Octal\n4. Hexadecimal\n");
    scanf("%d", &choice);
    switch(choice) {
        case 1:
            printf("Enter a binary number: ");
            scanf("%lld", &binaryNum);
            num = convertBinaryToDecimal(binaryNum);
            printf("Decimal: %d", num);
            decToBinary(num);
            decToOctal(num);
            decToHexa(num);
            break;
        case 2:
            printf("Decimal: %d", num);
            decToBinary(num);
            decToOctal(num);
            decToHexa(num);
            break;
        case 3:
            printf("Enter an octal number: ");
            scanf("%d", &octalNum);
            num = convertOctalToDecimal(octalNum);
            printf("Decimal: %d", num);
            decToBinary(num);
            decToOctal(num);
            decToHexa(num);
            break;
        case 4:

```

```

    printf("Enter a hexadecimal number: ");
    scanf("%s", hexNum);
    num = convertHexadecimalToDecimal(hexNum);
    printf("Decimal: %d", num);
    decToBinary(num);
    decToOctal(num);
    decToHexa(num);
    break;
default:
    printf("Invalid choice!");
    break;
}
return 0;
}

```

OUTPUT:

```

PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 6.c -o 6 } ; if ($?) { .\6 }
Enter the number: 10101
Choose the number system of the input number:
1. Binary
3. Octal
4. Hexadecimal
2
Decimal: 10101
Binary: 10011101110101
Octal: 23565
Hexadecimal: 2775
PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 6.c -o 6 } ; if ($?) { .\6 }
Choose the number system of the input number:
1. Binary
2. Decimal
3. Octal
4. Hexadecimal
3
Enter an octal number: 123457
Decimal: 42799
Binary: 1010011100101111
Octal: 123457
Hexadecimal: A72F
PS F:\C3> 

```

QUESTION 7:

Write a program to print all ASCII values [0-255] and the equivalent characters using a while loop.

SOLUTION:

```

#include <stdio.h>
int main()
{
    int i = 0;

```

```

printf("ASCII TABLE--\nINTEGER CHARACTER \n");
while (i < 256)
{
    printf("%d\t\t%c\n", i, (char)i);
    i++;
}

return 0;
}

```

OUTPUT:

```

PS F:\C3> cd "f:\C3\" ; if ($?) { gcc 7.c -o 7 } ; if ($?) { .\7 }
ASCII TABLE--
INTEGER CHARACTER
0
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     L
29     ⇄
30     ▲
31     ▼
32
33     !
34     "
35     #
36     $
37     %
38     &
39     '
40     (
41     )
42     *
43     +
44     ,
45     -
46     .
47     /
48     0
49     1
50     2
51     3
52     4
53     5
54     6
55     7
56     8
57     9
58     :
59     ;
60     <
61     =
62     >
63     ?
64     @
65     A
66     B
67     C
68     D
69     E
70     F
71     G
72     H
73     I
74     J
75     K
76     L
77     M
78     N
79     O
80     P
81     Q
82     R
83     S
84     T
85     U
86     V
87     W
88     X
89     Y
90     Z
91     [
92     \
93     ]
94     ^
95     _

```

