

Assignment 4 from UNIT - IV

AQ 1: WAP in C to find that whether a Integer number can be expressed as Sum of Two Prime Numbers

```
#include <stdio.h>
int checkPrime(int n);
int main()
{
    int n, i, flag = 0;

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

    for(i = 2; i <= n/2; ++i)
    {
        // condition for i to be a prime number
        if (checkPrime(i) == 1)
        {
            // condition for n-i to be a prime number
            if (checkPrime(n-i) == 1)
            {
                // n = primeNumber1 + primeNumber2
                printf("%d = %d + %d\n", n, i, n - i);
                flag = 1;
            }
        }
    }

    if (flag == 0)
        printf("%d cannot be expressed as the sum of two prime numbers.", n);

    return 0;
}

// Function to check prime number
int checkPrime(int n)
{
    int i, isPrime = 1;

    for(i = 2; i <= n/2; ++i)
    {
        if(n % i == 0)
        {
```

```
        isPrime = 0;
        break;
    }
}

return isPrime;
}
```

AQ 2: WAP to reverse a sentence entered by user without using string function by using recursion.

```
#include <stdio.h>
void reverseSentence();

int main()
{
    printf("Enter a sentence: ");
    reverseSentence();

    return 0;
}

void reverseSentence()
{
    char c;
    scanf("%c", &c);

    if( c != '\n')
    {
        reverseSentence();
        printf("%c",c);
    }
}
```

AQ 3: WAP to merge two files

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

int main()
{
    FILE *fs1, *fs2, *ft;

    char ch, file1[20], file2[20], file3[20];
```

MRSPTU ASSIGNMENT-4 COURSE CODE BCSE0-101

```
printf("Enter name of first file\n");
gets(file1);

printf("Enter name of second file\n");
gets(file2);

printf("Enter name of file which will store contents of two files\n");
gets(file3);

fs1 = fopen(file1,"r");
fs2 = fopen(file2,"r");

if( fs1 == NULL || fs2 == NULL )
{
    perror("Error ");
    printf("Press any key to exit...\n");
    getch();
    exit(EXIT_FAILURE);
}

ft = fopen(file3,"w");

if( ft == NULL )
{
    perror("Error ");
    printf("Press any key to exit...\n");
    exit(EXIT_FAILURE);
}

while( ( ch = fgetc(fs1) ) != EOF )
    fputc(ch,ft);

while( ( ch = fgetc(fs2) ) != EOF )
    fputc(ch,ft);

printf("Two files were merged into %s file successfully.\n",file3);

fclose(fs1);
fclose(fs2);
fclose(ft);

return 0;
}
```

AQ 4: WAP in C to add two complex numbers using structures.

```
#include <stdio.h>

struct complex
{
    int real, img;
};

int main()
{
    struct complex a, b, c;

    printf("Enter a and b where a + ib is the first complex number.\n");
    printf("a = ");
    scanf("%d", &a.real);
    printf("b = ");
    scanf("%d", &a.img);
    printf("Enter c and d where c + id is the second complex number.\n");
    printf("c = ");
    scanf("%d", &b.real);
    printf("d = ");
    scanf("%d", &b.img);

    c.real = a.real + b.real;
    c.img = a.img + b.img;

    if ( c.img >= 0 )
        printf("Sum of two complex numbers = %d + %di\n", c.real, c.img);
    else
        printf("Sum of two complex numbers = %d %di\n", c.real, c.img);

    return 0;
}
```

AQ 5: WAP in C to demonstrate the use of structure as function argument

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

struct Books {
    char title[50];
    char author[50];
    char subject[100];
}
```

```
int book_id;
};

/* function declaration */
void printBook( struct Books book );

int main( ) {

    struct Books Book1;    /* Declare Book1 of type Book */
    struct Books Book2;    /* Declare Book2 of type Book */

    /* book 1 specification */
    strcpy( Book1.title, "C Programming");
    strcpy( Book1.author, "Nuha Ali");
    strcpy( Book1.subject, "C Programming Tutorial");
    Book1.book_id = 6495407;

    /* book 2 specification */
    strcpy( Book2.title, "Telecom Billing");
    strcpy( Book2.author, "Zara Ali");
    strcpy( Book2.subject, "Telecom Billing Tutorial");
    Book2.book_id = 6495700;

    /* print Book1 info */
    printBook( Book1 );

    /* Print Book2 info */
    printBook( Book2 );

    return 0;
}

void printBook( struct Books book ) {

    printf( "Book title : %s\n", book.title);
    printf( "Book author : %s\n", book.author);
    printf( "Book subject : %s\n", book.subject);
    printf( "Book book_id : %d\n", book.book_id);
}
```

AQ 6: WAP in C to demonstrate the Dynamic Memory Allocation for Structure

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

MRSPTU ASSIGNMENT-4 COURSE CODE BCSE0-101

```
struct course
{
    int marks;
    char subject[30];
};

int main()
{
    struct course *ptr;
    int i, noOfRecords;
    printf("Enter number of records: ");
    scanf("%d", &noOfRecords);

    // Allocates the memory for noOfRecords structures with pointer ptr pointing to the base
    address.
    ptr = (struct course*) malloc (noOfRecords * sizeof(struct course));

    for(i = 0; i < noOfRecords; ++i)
    {
        printf("Enter name of the subject and marks respectively:\n");
        scanf("%s %d", &(ptr+i)->subject, &(ptr+i)->marks);
    }

    printf("Displaying Information:\n");

    for(i = 0; i < noOfRecords ; ++i)
        printf("%s\t%d\n", (ptr+i)->subject, (ptr+i)->marks);

    return 0;
}
```