



BMS COLLEGE OF ENGINEERING, BENGALURU-560019

(Autonomous Institute, Affiliated to VTU)

Academic Year- 2022-2023

Course: Principles of Programming using C

Code: 22CS1ESPOP

Part B Programs

1. Develop a C program to search a Book ID from an organized bookshelf which has N number of Books using appropriate searching technique.

Program:

```
#include <stdio.h>
int main()
{
    int i, low, high, mid, n, key, array[100];

    printf("Enter number of books in the shelf\n");
    scanf("%d",&n);

    printf("Enter %d BookIds\n", n);
    for(i = 0; i < n; i++)
        scanf("%d",&array[i]);

    printf("Enter BookID(number) to find\n");
    scanf("%d", &key);

    low = 0;
    high = n - 1;
    mid = (low+high)/2;

    while (low <= high)
    {
        if(array[mid] < key)
            low = mid + 1;
        else if (array[mid] == key)
        {
            printf("BookId=%d is found at location %d\n", key, mid+1);
            break;
        }
        else
            high = mid - 1;

        mid = (low + high)/2;
    }

    if(low > high)
```

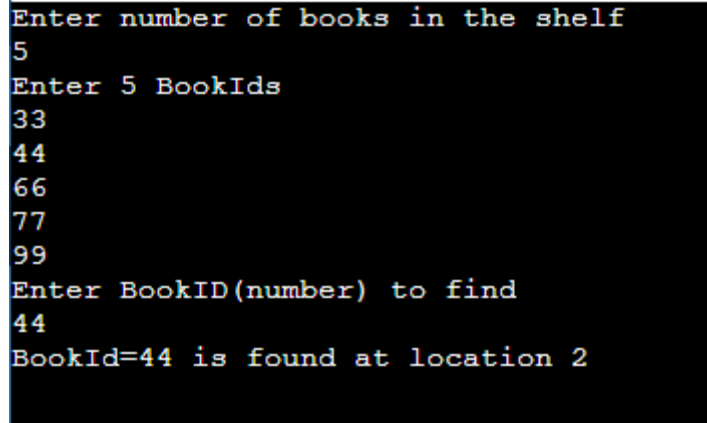
```

        printf("BookID Not found! %d isn't present in the Book Shelf\n", key);

    return 0;
}

```

Output:



```

Enter number of books in the shelf
5
Enter 5 BookIds
33
44
66
77
99
Enter BookID(number) to find
44
BookId=44 is found at location 2

```

2. Develop a C Program to find the Transpose a Matrix.

Program:

```

#include <stdio.h>
int main()
{
    int m, n, i, j, matrix[10][10], transpose[10][10];

    printf("Enter rows and columns :\n");
    scanf("%d%d", &m, &n);

    printf("Enter elements of the matrix\n");
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &matrix[i][j]);

    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            transpose[j][i] = matrix[i][j];

    printf("Transpose of the matrix:\n");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
            printf("%d\t", transpose[i][j]);
        printf("\n");
    }
    return 0;
}

```

Output:

```

Enter rows and columns:
2 2

```

Enter elements of the matrix

1 2

3 4

Transpose of the matrix:

1 3

2 4

3. Write functions to implement String operations such as concatenate and find String length without using built-in functions.

Program:

```
#include <stdio.h>
void concatenate(char [], char []);
int string_length(char []);

int main()
{
    char s1[100], s2[100];

    printf("Input a string1\n");
    gets(s1);

    printf("Input a string2\n");
    gets(s2);

    int length1 = string_length(s1);
    int length2 = string_length(s2);
    printf("Length of %s = %d\n", s1, length1);
    printf("Length of %s = %d\n", s2, length2);

    concatenate(s1, s2);
    printf("String obtained on concatenation: \"%s\"\n", s1);

    return 0;
}

int string_length(char s1[])
{
    int i = 0;
    while (s1[i] != '\0')
    {
        i++;
    }
    return i;
}

void concatenate(char s1[], char s2[])
{
    int i=0, j=0;
    while (s1[i] != '\0')
    {
        i++;
    }
    while (s2[j] != '\0')
    {
```

```

        s1[i] = s2[j];
        j++;
        i++;
    }
    s1[i] = '\0';
}

```

Output:

```

Input a string1
hi
Input a string2
hello
Length of hi = 2
Length of hello = 5
String obtained on concatenation: "hihello"

```

4. Parameter Passing techniques: Call by Value and Call by reference (Virtual Lab link: <https://cse02-iiith.vlabs.ac.in/exp/pointers/>)

Develop a C program to swap two integer numbers using call by value and call by reference parameter passing technique.

```

#include <stdio.h>

void swap_call_by_val(int x,int y);
void swap_call_by_ref(int *ptrx,int *ptr);

int main()
{
    int x, y,mode;
    printf("Enter Value of x ");
    scanf("%d", &x);
    printf("\nEnter Value of y ");
    scanf("%d", &y);

    swap_call_by_val(x,y);
    swap_call_by_ref(&x,&y);
    return 0;
}

void swap_call_by_val(int x,int y)
{
    int temp = x;
    x = y;
    y = temp;
    printf("\nAfter Swapping using call by value: x = %d, y = %d", x, y);
}

void swap_call_by_ref(int *ptrx,int *ptr)
{
    int temp = *ptrx;
    *ptrx = *ptr;
    *ptr = temp;
}

```

```

    printf("\nAfter Swapping using call by reference: x = %d, y = %d", *ptrx, *ptry);
}

```

Output:

```

Enter Value of x 4

Enter Value of y 5

After Swapping using call by value: x = 5, y = 4
After Swapping using call by reference: x = 5, y = 4

```

5. Structures (Virtual Lab link: <https://cse02-iiith.vlabs.ac.in/exp/structures/>)

Implement structures to read, write and compute the average salary for a department of N employees. (Consider nested structure for DOB)

Program:

```

#include<stdio.h>
struct dateofBirth{
    int dd;
    int mm;
    int yy;
};

struct employee
{
    int eno;
    char ename[20];
    struct dateofBirth DOB;
    int salary;
}emp[10];

int main ()
{
    int i, high, n, sal_sum = 0, sal_avg = 0;
    printf ("How many employee info\nyou want to accept: ");
    printf ("Enter Limit: ");
    scanf ("%d", &n);
    printf ("Enter details for %d employees:", n);
    for (i = 0; i < n; i++)
    {
        printf ("Employee Number: ");
        scanf ("%d", &emp[i].eno);
        printf ("Name          : ");
        scanf ("%s", emp[i].ename);
        printf("Enter Date of Birth [DD MM YYYY] format: ");
        scanf("%d%d%d",&emp[i].DOB.dd,&emp[i].DOB.mm,&emp[i].DOB.yy);
        printf ("Salary          : ");
        scanf ("\n %d", &emp[i].salary);
    }
}

```

```

    for (i = 0; i < n; i++)
    {
        sal_sum = sal_sum + emp[i].salary;
    }
    sal_avg = sal_sum / n;
    printf ("The average salary of the employees is %d\n", sal_avg);

    return 0;
}

```

Output:

```

How many employee info
you want to accept: Enter Limit: 3
-----
Enter details for 3 employees:
-----
Employee Number: 2
Name           : gg
Enter Date of Birth [DD MM YYYY] format: 22 11 2002
Salary         : 3333
-----
Employee Number: 3
Name           : ff
Enter Date of Birth [DD MM YYYY] format: 11 09 2000
Salary         : 2222
-----
Employee Number: 1
Name           : ss
Enter Date of Birth [DD MM YYYY] format: 11 02 1998
Salary         : 8888
-----
The average salary of the employees is 4814

```

6. Demonstrate how to read data from the keyboard, write it to a file called BMSCE, again read the same data from the BMSCE file, and display it on the screen/console.

Program:

```

#include <stdio.h>
int main()
{
    FILE *fp; char c;
    printf("Data read from the keyboard : Input\n\n");

    /* Open the file INPUT */
    fp = fopen("BMSCE.txt", "a");

    /* Get a character from keyboard */
    while((c=getchar()) != EOF)
        /* Write a character to INPUT */
        fputc(c,fp);

    /* Close the file INPUT */
    fclose(fp);

    printf("\nData Output\n\n");
    /* Reopen the file INPUT */
    fp = fopen("BMSCE.txt", "r");
}

```

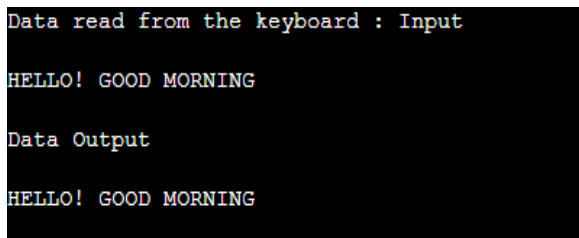
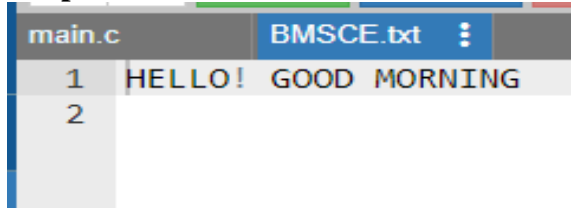
```

/* Read a character from INPUT*/
while((c=fgetc(fp)) != EOF)
    /* Display a character on screen */
    printf("%c",c);

/* Close the file INPUT */
fclose(fp);
}

```

Output:



Steps for executing file program in codeblocks:

1. First **build** the program then **run**.
2. Type some text when you see “**Input into file**” on output screen.
3. Then press **enter** key, then press **ctrl+z**, then again press **enter** key.

Steps for executing file program in onlineGDB compiler:

1. First **run** the program.
2. Type some text when you see “**Input into file**” on output screen.
3. Then press **enter** key, then press **ctrl+d**.

Note: This program does not work in other online c compilers. So, use only onlineGDB compiler.