



SYMBIOSIS INSTITUTE OF TECHNOLOGY (SIT)

Constituent of Symbiosis International (Deemed University), Pune

(Established under Section 3 of the UGC Act of 1956 vide notification number F-9-12/2001-U-3 of the Government of India)

Re-Accredited by NAAC with 'A++' Grade

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Programming in C

Lab Journal

Name: Faheemuddin Sayyed

Batch: CSE C-1

PRN: 23070122196

SEM-II

A.Y. 2023-2024

List of Experiments

| Sr. No. | Title of Experiment |
|--------------------|--|
| 1 | Design and develop a flowchart or an algorithm that takes three coefficients a,b and c of a quadratic equation ($ax^2+bx+c=0$) as input and compute all possible roots. |
| 2 | Write a C program for the developed flowchart/algorithim and execute the same to output the possible roots for a given set of coefficients with appropriate messages. |
| 3 | Write a C Program to relate two integers using =, > or < using nested ifs & multiple ifs. |
| 4 | Write a C program to find whether (using switch case) <ul style="list-style-type: none"> (a) A given number is prime or composite. (b) A given number is even or odd. |
| 5 | Write a C program to generate the Fibonacci Series for a given number ‘n’. |
| 6 | Write a C program of binary search for a user-given element in a set of ‘n’ numbers. |
| 7 | Write a C program to perform matrix operations like addition, subtraction and transpose |
| 8 | Write a C program to accept a string from console and to display the following on console (without using built-in functions): <ul style="list-style-type: none"> (a) Length of the string (b) Total number of characters in the string (c) Total number of vowels in the string (d) Copy one string into the other. |
| 9 | Write a C program using functions to accept two strings from the console and perform the following operations (without using built-in functions): <ul style="list-style-type: none"> (a) Compare the strings: equal/not equal (b) Find the longer string. (c) Concatenate the two strings. (d) Find occurrence of substring in the main string |
| 10 | Write a C program using functions to accept a one-dimensional array of integers and sort them in ascending order. |

| | |
|----|---|
| 11 | Write a C program of selection sorting using functions. |
| 12 | Write a C program to find the factorial of a number using recursion |
| 13 | Write a C program using structures to print the pay slip of an employee after accepting details like id. no, name, designation, department and basic salary |
| 14 | <ol style="list-style-type: none"> 1. Write a program to demonstrate compile time memory allocation 2. Write a program to demonstrate runtime memory allocation |
| 15 | Write a C Program to demonstrate Structure pointer. |

Prepared By:

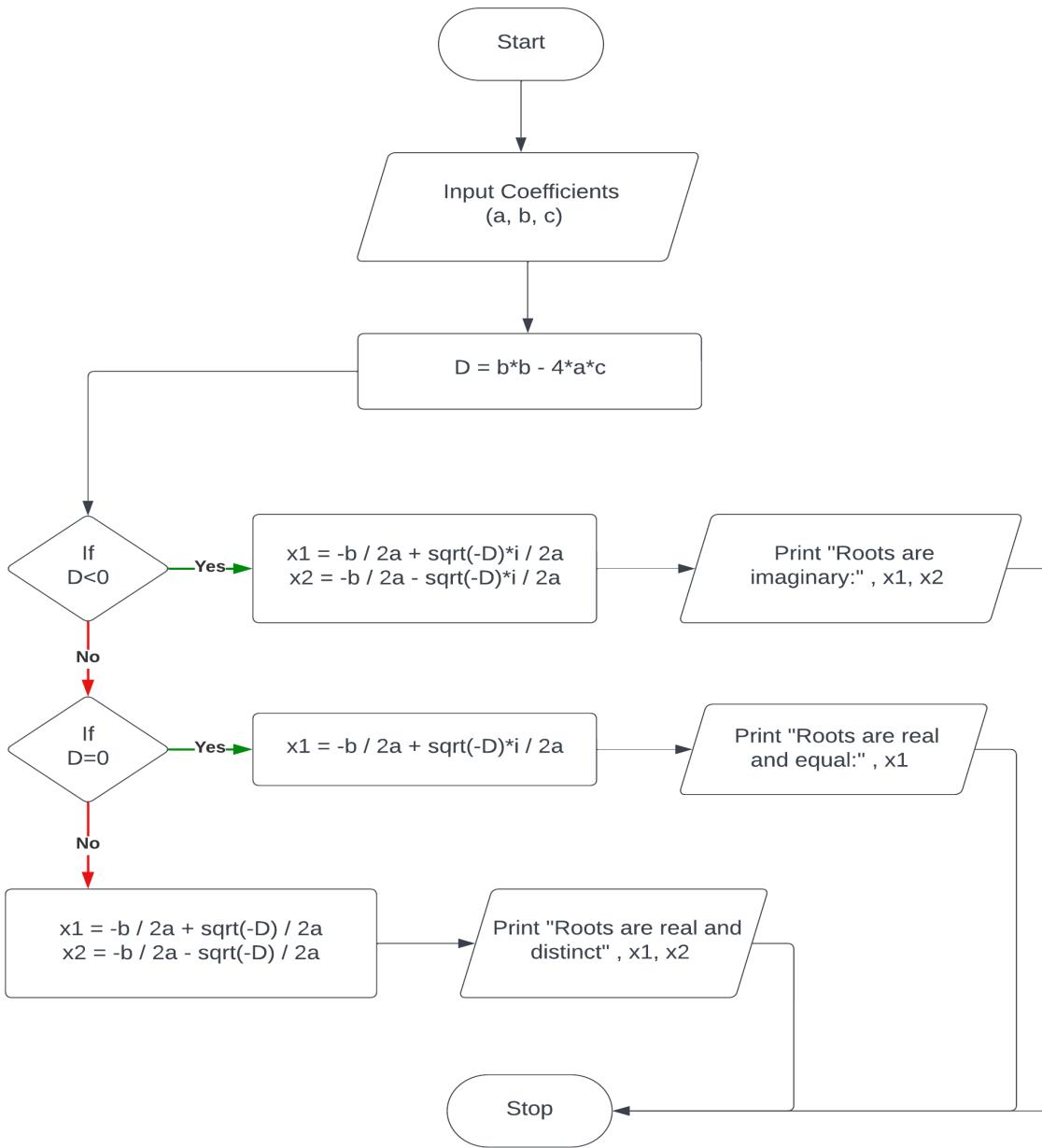
Faculty Name(s): Mr. Mahesh Arse, Dr. Kanhaiya Sharma, Mr. Sachin Gaikwad



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 01 | |
|---|---|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 12/01/2024 |
| Assignment Title (Full): | Design and develop a flowchart or an algorithm that takes three coefficients a, b and c of a quadratic equation ($ax^2+bx+c=0$) as input and compute all possible roots. |
| Theory: | (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) |
| Theory Response: | |
| 1. Accept coefficients a, b, and c as input. 2. Calculate the discriminant (Δ) using the formula $\Delta = b^2 - 4ac$. 3. Check the value of Δ : <ul style="list-style-type: none">o If Δ is positive, calculate two roots using the quadratic formula:<ul style="list-style-type: none">▪ $root1 = (-b + \sqrt{\Delta}) / (2a)$▪ $root2 = (-b - \sqrt{\Delta}) / (2a)$o If Δ is zero, calculate a single root using the formula:<ul style="list-style-type: none">▪ $root = -b / (2a)$o If Δ is negative, the roots are imaginary and are calculated as:<ul style="list-style-type: none">▪ $root1 = (-b + \sqrt{-\Delta}) * i / (2a)$▪ $root2 = (-b - \sqrt{-\Delta}) * i / (2a)$[Here, i is iota, i.e. $\sqrt{-1}$] | |
| 4. Display the roots. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |
| Output Response: | |



Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

The flowchart calculates the roots of a quadratic equation based on the input coefficients. By considering the discriminant, the program determines the nature and number of roots and computes them accordingly.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
 Prof. Mahesh Arse
 Prof. Sachin R. Gaikwad
 Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 02 | |
|---|--|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 19/01/2024 |
| Assignment Title (Full): | Write a C program for the developed flowchart/algorithm and execute the same to output the possible roots for a given set of coefficients with appropriate messages. |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithm of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: | |
| <ol style="list-style-type: none">1. User Input: Get coefficients (a, b, c) for a quadratic equation from the user.2. Calculate Discriminant (D): Compute the discriminant ($D = b^2 - 4ac$).3. Check Discriminant:<ul style="list-style-type: none">o If $D > 0$, calculate two real and different roots using the quadratic formula.o If $D = 0$, calculate two real and identical roots.o If $D < 0$, calculate two complex roots.4. Display Results: Print the type of roots and their values. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |

Output Response:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     double a, b, c, D, x1, x2;
6     printf("Enter coefficients a, b, and c: ");
7     scanf("%lf %lf %lf", &a, &b, &c);
8     D = b * b - 4 * a * c;
9     switch (D > 0) {
10         case 1:
11             x1 = (-b + sqrt(D)) / (2 * a);
12             x2 = (-b - sqrt(D)) / (2 * a);
13             printf("Roots are real and different.\n");
14             printf("Root 1 = %.2lf\n", x1);
15             printf("Root 2 = %.2lf\n", x2);
16             break;
17         case 0:
18             switch (D == 0) {
19                 case 1:
20                     x1 = x2 = -b / (2 * a);
21                     printf("Roots are real and same.\n");
22                     printf("Root 1 = Root 2 = %.2lf\n", x1);
23                     break;
24                 case 0:
25                     printf("Roots are complex and different.\n");
26                     double realPart = -b / (2 * a);
27                     double imaginaryPart = sqrt(-D) / (2 * a);
28                     printf("Root 1 = %.2lf + %.2lfi\n", realPart, imaginaryPart);
29                     printf("Root 2 = %.2lf - %.2lfi\n", realPart, imaginaryPart);
30                     break;
31             }
32             break;
33     }
34     return 0;
35 }
```

```
Enter coefficients a, b, and c: 1 4 1
Roots are real and different.
```

```
Root 1 = -0.27
```

```
Root 2 = -3.73
```

```
› (base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

```
Enter coefficients a, b, and c: 1 2 1
```

```
Roots are real and same.
```

```
Root 1 = Root 2 = -1.00
```

```
○ (base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

```
Enter coefficients a, b, and c: 1 2 2
```

```
Roots are complex and different.
```

```
Root 1 = -1.00 + 1.00i
```

```
Root 2 = -1.00 - 1.00i
```

```
› (base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

In conclusion, the provided C code takes user input for quadratic equation coefficients, calculates the discriminant, and determines the nature of the roots (real and different, real and identical, or complex). The program then displays the type of roots and their respective values. The code utilizes switch statements to handle different cases based on the discriminant's value.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 03 | |
|--|---|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 02/02/2024 |
| Assignment Title (Full): | Write a C Program to relate two integers using =, > or < using nested ifs & multiple ifs. |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: <ol style="list-style-type: none">Accept two integers (num1 and num2) as user input.Use the ternary conditional operator, switch case statement and if-else conditional statements to compare the numbers and print the result:<ul style="list-style-type: none">If num1 equals num2, print "Both numbers are equal."If num1 is greater than num2, print "num1 is greater than num2."If num2 is greater than num1, print "num2 is greater than num1." | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |
| Output Response: | |

```

1 #include <stdio.h>
2
3 int main() {
4     int num1, num2;
5     printf("Enter the first number: ");
6     scanf("%d", &num1);
7     printf("Enter the second number: ");
8     scanf("%d", &num2);
9
10    (num1 == num2)?printf("Both numbers are equal.\n"):(num1 > num2)?printf("%d is \
11 greater than %d.\n", num1, num2):printf("%d is greater than %d.\n", num2, num1);
12 /*
13     switch (num1 > num2)
14     {
15     case 1:
16         printf("%d is greater than %d.\n", num1,num2);
17         break;
18     case 0:
19         switch (num1<num2)
20         {
21         case 1:
22             printf("%d is greater than %d.\n", num2,num1);
23             break;
24
25         case 0:
26             printf("Both numbers are equal.\n");
27             break;
28         }
29     }
30 */
31 /*
32     if (num1 == num2) {
33         printf("Both numbers are equal.\n");
34     }
35     else if (num1 > num2) {
36         printf("%d is greater than %d.\n",num1,num2);
37     }
38     else {
39         printf("%d is greater than %d.\n",num2,num1);
40     }
41 */
42
43     return 0;
44 }

```

Enter the first number: 12
Enter the second number: 15
15 is greater than 12.

› (base) fahee@Faheems-MacBook-Pro Programming_in_C %

Enter the first number: 15
Enter the second number: 9
15 is greater than 9.

› (base) fahee@Faheems-MacBook-Pro Programming_in_C %

```
Enter the first number: 12
Enter the second number: 12
Both numbers are equal.
> (base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

The C code compares two numbers using a ternary conditional operator, determining whether they are equal or which one is greater. The commented-out sections include alternative implementations using switch statements and if-else conditions, showcasing different ways to achieve the same result.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 04 | |
|---|--|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 09/02/2024 |
| Assignment Title (Full): | Write a C program to find whether (using switch case) (a) A given number is prime or composite. (b) A given number is even or odd. |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: <ol style="list-style-type: none">Prompt the user to choose between checking if a number is prime/composite or even/odd.Accept user input for the choice (n) and the number (num).Use a nested switch statement based on the user's choice (n):<ul style="list-style-type: none">Case 1: Check if the number is prime or composite.Case 2: Check if the number is even or odd.Default: Print an error message for an invalid choice.For prime or composite:<ul style="list-style-type: none">Handle special cases (0, 1, 2).Use a loop to check divisibility and determine if the number is prime or composite.For even or odd:<ul style="list-style-type: none">Use the modulo operator to check parity and print the result. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |

Output Response:

```
1 #include <stdio.h>
2
3 int main() {
4     int num,n;
5     int isPrime = 1;
6     printf("\nEnter accordingly:\n1. Prime or Composite\n2. Even or Odd\n");
7     scanf("%d",&n);
8     printf("\nEnter a number: ");
9     scanf("%d", &num);
10
11    switch (n) {
12        case 1:
13            switch (num) {
14                case 0:
15                case 1:
16                    printf("%d is neither prime nor composite.\n", num);
17                    break;
18                case 2:
19                    printf("%d is a prime number.\n", num);
20                    break;
21                default:
22                    for (int i = 2; i < num; i++) {
23                        if (num % i == 0) {
24                            isPrime = 0;
25                            break;
26                        }
27                    }
28                    if (isPrime) {
29                        printf("%d is a prime number.\n", num);
30                    } else {
31                        printf("%d is a composite number.\n", num);
32                    }
33                    break;
34                }
35                break;
36            case 2:
37                switch (num % 2) {
38                    case 0:
39                        printf("%d is an even number.\n", num);
40                        break;
41
42                    case 1:
43                        printf("%d is an odd number.\n", num);
44                        break;
45                }
46                default:
47                    printf("\nPlease enter a valid number.\n");
48                    break;
49            }
50
51    return 0;
52 }
```

```
Enter accordingly:  
1. Prime or Composite  
2. Even or Odd  
1
```

```
Enter a number: 7  
7 is a prime number.
```

```
○ (base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

```
Enter accordingly:  
1. Prime or Composite  
2. Even or Odd  
1
```

```
Enter a number: 8  
8 is a composite number.
```

```
○ (base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

```
Enter accordingly:  
1. Prime or Composite  
2. Even or Odd  
2
```

```
Enter a number: 10  
10 is an even number.
```

```
○ (base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

```
Enter accordingly:  
1. Prime or Composite  
2. Even or Odd  
2
```

```
Enter a number: 11  
11 is an odd number.
```

```
○ (base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

The C code offers the user the choice to check whether a given number is prime or composite or if it is even or odd. It employs a switch statement to efficiently navigate between these options, incorporating loops and conditional checks for accurate results.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 05 | |
|---|--|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 09/02/2024 |
| Assignment Title (Full): | Write a C program to generate the Fibonacci Series for a given number 'n'. |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: <ol style="list-style-type: none">Accept the number of terms (n) for the Fibonacci series from the user.Declare an array A of size n to store the Fibonacci series.Initialize the first two elements of the array as 0 and 1.Use a loop to calculate the remaining Fibonacci numbers and store them in the array.Display the generated Fibonacci series using another loop. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |

Output Response:

```
1 # include <stdio.h>
2 int main(){
3     int n;
4     printf("Enter the number of terms: ");
5     scanf("%d", &n);
6     int A[n];
7     A[0]=0;
8     A[1]=1;
9     for (int i = 2; i < n; i++)
10    {
11        A[i]=A[i-1]+A[i-2];
12    }
13    printf("The fibonacci series is: ");
14    for (int i = 0; i < n; i++)
15    {
16        printf("%d ", A[i]);
17    }
18 }
```

```
Enter the number of terms: 11
The fibonacci series is: 0 1 1 2 3 5 8 13 21 34 55
(base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

The C code generates the Fibonacci series up to the specified number of terms. It efficiently utilizes an array to store and compute the series. The Fibonacci sequence is then displayed to the user. The code showcases a simple and effective approach to generating and presenting the Fibonacci series.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 06 | |
|--|--|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 16/02/2024 |
| Assignment Title (Full): | Write a C program of binary search for a user-given element in a set of 'n' numbers. |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: | |
| <ol style="list-style-type: none">1. Accept the size of the array (size) and allocate memory dynamically.2. Input sorted elements into the array.3. Accept the element to search (x).4. Implement a binary search function (BinarySearch) to find the index of the element in the array.5. Display whether the element is found and its index. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |

Output Response:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int BinarySearch(int arr[], int length, int x){
5     int l=0, h=length-1;
6     while (l<=h){
7         int mid = (l+h)/2;
8         if (arr[mid] == x){
9             return mid;
10        } else if (arr[mid] < x){
11            l = mid+1;
12        } else {
13            h = mid-1;
14        }
15    }
16    return -1;
17 }
18
19 int main(){
20     int size,*arr,x;
21     printf("\nEnter the size of the array: ");
22     scanf("%d", &size);
23     arr = (int *)malloc(size * sizeof(int));
24     printf("\nEnter the elements of the array (sorted): ");
25     for (int i=0; i<size; i++){
26         scanf("%d", &arr[i]);
27     }
28     printf("\nEnter the element to search: ");
29     scanf("%d", &x);
30     int result = BinarySearch(arr, size, x);
31     if (result == -1){
32         printf("Element not found.\n");
33     } else {
34         printf("Element found at index %d.\n", result);
35     }
36     free(arr);
37     return 0;
38 }
```

```
Enter the size of the array: 10
Enter the elements of the array (sorted): 1 2 3 4 5 6 7 8 9 10
Enter the element to search: 6
Element found at index 5.
(base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

```
Enter the size of the array: 10
Enter the elements of the array (sorted): 1 2 3 4 5 6 7 8 9 10
Enter the element to search: 11
Element not found.
(base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

The C code demonstrates a binary search on a sorted array. It dynamically allocates memory for the array, takes user input for sorted elements, and performs a binary search for a specified element. The code efficiently handles memory allocation and deallocation while providing a clear and functional binary search implementation. The result informs the user about whether the element is found and, if so, at which index in the array.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 07 | |
|---|--|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 23/02/2024 |
| Assignment Title (Full): | Write a C program to perform matrix operations like addition, subtraction and transpose. |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: 1. Declare matrices 'a', 'b', and 'c'. 2. Use nested loops to add corresponding elements of matrices 'a' and 'b' and store the result in matrix 'c'. 3. Print matrices 'a' and 'b'. 4. Print the sum of matrices 'a' and 'b' (matrix 'c'). 5. Print the difference of matrices 'a' and 'b'. 6. Print the transpose of matrix 'a'. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |

Output Response:

```
1 #include <stdio.h>
2
3 int main(){
4     int a[2][2] = {{1,2},{3,4}};
5     int b[2][2] = {{5,6},{7,8}};
6     int c[2][2];
7     for (int i=0; i<2; i++){
8         for (int j=0; j<2; j++){
9             c[i][j] = a[i][j] + b[i][j];
10        }
11    }
12    printf("\nMatrix A:\n");
13    for (int i=0; i<2; i++){
14        for (int j=0; j<2; j++){
15            printf("%d ", a[i][j]);
16        }
17        printf("\n");
18    }
19    printf("\nMatrix B:\n");
20    for (int i=0; i<2; i++){
21        for (int j=0; j<2; j++){
22            printf("%d ", b[i][j]);
23        }
24        printf("\n");
25    }
26
27 // Addition of matrices:
28 printf("\nMatrix C (A+B):\n");
29 for (int i=0; i<2; i++){
30     for (int j=0; j<2; j++){
31         printf("%d ", c[i][j]);
32     }
33     printf("\n");
34 }
35
36 // Subtraction of matrices:
37 printf("\nMatrix C (A-B):\n");
38 for (int i=0; i<2; i++){
39     for (int j=0; j<2; j++){
40         printf("%d ", a[i][j]-b[i][j]);
41     }
42     printf("\n");
43 }
44
45 // Transpose of a matrix:
46 printf("\nMatrix A (Transpose):\n");
47 for (int i=0; i<2; i++){
48     for (int j=0; j<2; j++){
49         printf("%d ", a[j][i]);
50     }
51     printf("\n");
52 }
53
54 }
```

```
Matrix A:  
1 2  
3 4  
  
Matrix B:  
5 6  
7 8  
  
Matrix C (A+B):  
6 8  
10 12
```

```
Matrix C (A-B):  
-4 -4  
-4 -4
```

```
Matrix A (Transpose):  
1 3  
2 4
```

```
> (base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

The code performs basic matrix operations - addition, subtraction, and transpose. It initializes matrices 'a' and 'b', computes the sum and difference, and prints the original matrices along with the results of these operations. The transpose of matrix 'a' is also calculated and displayed. The code demonstrates fundamental matrix manipulation in C.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 08 | |
|---|---|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 29/03/2024 |
| Assignment Title (Full): | <p>Write a C program to accept a string from console and to display the following on console (without using built-in functions):</p> <p>(a) Length of the string (b) Total number of characters in the string (c) Total number of vowels in the string (d) Copy one string into another</p> |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: | |
| <ol style="list-style-type: none">1) Define functions for calculating string length, converting characters to uppercase, counting characters, counting vowels, and copying strings.2) Implement the length function to count characters in a string.3) Implement Upper function to convert characters to uppercase.4) Implement char_len function to count characters in a string (ignores non-alphabetic characters).5) Implement vow_len function to count vowels in a string (considers both uppercase and lowercase).6) Implement Copy function to copy one string to another.7) In the main function, declare two arrays to store strings.8) Read a string from the user.9) Copy the string to another array.10) Print the copied string, length of the string, number of characters, and number of vowels. | |

Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.)

Output Response:

```
1 #include <stdio.h>
2
3 int length(char str[50]){
4     int count=0;
5     for(;str[count]!=0;count++);
6     return count;
7 }
8
9 void Upper(char str[50]){
10    for(int i=0;i<length(str);i++){
11        if(str[i]>=97 && str[i]<=122){
12            str[i]-=32;
13        }
14    }
15 }
16
17 int char_len(char str[50]){
18    int count=0;
19    for(int i=0;str[i]!=0;i++){
20        if((str[i]>=65 && str[i]<=90) || (str[i]>=97 && str[i]<=122)) count++;
21    }
22    return count;
23 }
24
25 int vow_len(char str[50]){
26    int count=0;
27    Upper(str);
28    for(int i=0;str[i]!=0;i++){
29        if(str[i]=='A' || str[i]=='E' || str[i]=='I' || str[i]=='O' || str[i]=='U') count++;
30    }
31    return count;
32 }
33
34 void Copy(char str1[50],char str2[50]){
35    for(int i=0;i<length(str1);i++) str1[i]=0;
36    for(int i=0;i<length(str2);i++){
37        str1[i]=str2[i];
38        str1[length(str2)] = '\0';
39    }
40 }
41
42 int main(){
43     char c1[50],c2[50];
44     printf("\nEnter string:\n");
45     scanf("%[^\\n]s",c2);
46     Copy(c1,c2);
47     printf("\nCopied string: %s\n",c1);
48     printf("\nLength of string: %d\n",length(c2));
49     printf("\nNo. of characters: %d\n",char_len(c2));
50     printf("\nNo. of vowels: %d\n",vow_len(c2));
51     return 0;
52 }
```

```
Enter string:  
Hello World!
```

```
Copied string: Hello World!
```

```
Length of string: 12
```

```
No. of characters: 10
```

```
No. of vowels: 3
```

```
› (base) faheem@Faheems-MacBook-Pro Programming_in_C %
```

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

The program defines functions to manipulate strings and provides basic functionalities such as copying strings, counting characters, and counting vowels. It demonstrates the use of functions to perform these operations on user-input strings.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 09 | |
|---|---|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 05/04/2024 |
| Assignment Title (Full): | <p>Write a C program using functions to accept two strings from the console and perform the following operations (without using built-in functions):</p> <p>(a) Compare the strings: equal/not equal (b) Find the longer string. (c) Concatenate the two strings. (d) Find occurrence of substring in the main string</p> |
| Theory: | (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) |
| Theory Response: | |
| <ul style="list-style-type: none">• Function Definitions:<ol style="list-style-type: none">1. length: Calculates the length of a string.2. Compare: Compares two strings for equality.3. Longer: Checks which string is longer.4. Concatenate: Concatenates two strings.5. Substring: Checks if one string contains another as a substring. | |
| <ul style="list-style-type: none">• Main Function:<ol style="list-style-type: none">1. Input: Reads two strings (s1 and s2) and a substring (sub) from the user.2. Comparison: Checks if s1 and s2 are equal using Compare.3. Length Comparison: Determines which string (s1 or s2) is longer using Longer.4. Concatenation: Concatenates s2 onto s1 using Concatenate.5. Substring Check: Determines if s2 contains sub as a substring using Substring. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |

Output Response:

```
1  #include <stdio.h>
2
3  int length(char str[50]){
4      int count=0;
5      for(;str[count]!=0;count++);
6      return count;
7  }
8
9  int Compare(char s1[50],char s2[50]){
10     if(length(s1)!=length(s2)) return 0;
11     for(int i=0;i<length(s1);i++){
12         if(s1[i]!=s2[i]) return 0;
13     }
14     return 1;
15 }
16
17 int Longer(char s1[50], char s2[50]){
18     if(length(s1)>length(s2)) return 1;
19     return 0;
20 }
21
22 void Concatenate(char s1[50], char s2[50]){
23     int l1=0,l2=0;
24     l1=length(s1);
25     l2=length(s2);
26     for(int i=l1,j=0;i<l1+l2 && j<l2;i++,j++){
27         s1[i]=s2[j];
28     }
29 }
30
31 int Substring(char s1[50], char sub[50]){
32     for(int i=0;i<length(s1);i++){
33         for(int i=0;i<=length(s1)-length(sub);i++){
34             int j;
35             for(j=0;j<length(sub);j++){
36                 if(s1[i+j]!=sub[j]){
37                     break;
38                 }
39             }
40             if(j==length(sub)) return 1;
41         }
42         return 0;
43     }
44     return 0;
45 }
46
47 int main(){
48     char s1[50], s2[50],sub[50];
49     printf("\nEnter 1st string: ");
50     scanf("%[^\\n]s",s1);
51     getchar(); // Consume the newline character
52     printf("\nEnter 2nd string: ");
53     scanf("%[^\\n]s",s2);
54     getchar();
55     Compare(s1,s2)?printf("\nStrings are equal.\n"):printf("\nStrings are not equal.\n");
56     Longer(s1,s2)?printf("\n1st String is Longer.\n"):printf("\n2nd String is Longer.\n");
57     Concatenate(s1,s2);
58     printf("\nConcatenated string: %s\n",s1);
59     printf("\nEnter substring to check in string 2:");
60     scanf("%[^\\n]s",sub);
61     Substring(s2,sub)?printf("\nContains Substring.\n"):printf("\nDoes not contain substring.\n");
62     return 0;
63 }
```

Enter 1st string: Hello World!

Enter 2nd string: Welcome to C

Strings are not equal.

2nd String is Longer.

Concatenated string: Hello World!Welcome to C

Enter substring to check in string 2:Welcome

Contains Substring.

○ (base) fahee@Faheems-MacBook-Pro Programming_in_C %

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

This program demonstrates basic string manipulation operations such as length calculation, comparison, concatenation, and substring checking using user-provided strings. It utilizes custom functions for these operations to encapsulate common tasks, promoting code reusability and clarity. However, it assumes string lengths won't exceed a certain limit (50 characters), which can be a limitation in practical scenarios.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma

Prof. Mahesh Arse

Prof. Sachin R. Gaikwad

Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 10 | |
|---|---|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 05/04/2024 |
| Assignment Title (Full): | Write a C program using functions to accept a one-dimensional array of integers and sort them in ascending order. |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: <ul style="list-style-type: none">Define a swap function that exchanges the values of two integers using pointers.Implement bubbleSort function that iterates through the array multiple times, comparing adjacent elements and swapping them if they are in the wrong order.<ul style="list-style-type: none">Outer loop controls the number of passes through the array.Inner loop compares adjacent elements and swaps them if necessary.In main, initialize an integer array.Determine the number of elements in the array.Call bubbleSort passing the array and its size.Output the sorted array using printf. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |

Output Response:

```
1 #include <stdio.h>
2
3 void swap(int *a, int *b) {
4     int temp = *a;
5     *a = *b;
6     *b = temp;
7 }
8
9 void bubbleSort(int arr[], int n) {
10    for (int i = 0; i < n-1; i++) {
11        for (int j = 0; j < n-i-1; j++) {
12            if (arr[j] > arr[j+1]) {
13                swap(&arr[j], &arr[j+1]);
14            }
15        }
16    }
17 }
18
19 int main() {
20     int arr[] = {64, 34, 25, 12, 22, 11, 90};
21     int n = sizeof(arr)/sizeof(arr[0]);
22     bubbleSort(arr, n);
23     printf("\nSorted array: \n");
24     for (int i = 0; i < n; i++) {
25         printf("%d ", arr[i]);
26     }
27     printf("\n");
28     return 0;
29 }
```

Sorted array:

11 12 22 25 34 64 90

› (base) fahee@Faheems-MacBook-Pro Programming_in_C %

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

The code sorts an integer array in ascending order using the bubble sort algorithm and demonstrates the use of function calls and array manipulation in C. Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. This process is repeated until the array is fully sorted.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 11 | |
|---|---|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 12/04/2024 |
| Assignment Title (Full): | Write a C program of selection sorting using functions. |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: | |
| <ul style="list-style-type: none">• SelectionSort Function:<ol style="list-style-type: none">1. Initialize min_index to track the minimum element index.2. Iterate through the array from i = 0 to n-1.3. Find the minimum element in the unsorted portion of the array.4. Swap the minimum element with the element at i if necessary. | |
| <ul style="list-style-type: none">• Main Function:<ol style="list-style-type: none">1. Declare and initialize an array arr.2. Determine the size n of the array.3. Call SelectionSort function with arr and n.4. Print the sorted array using a for loop.5. Return 0 to indicate successful completion. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |

Output Response:

```
19 #include <stdio.h>
20 void SelectionSort(int arr[],int n){
21     int min_index=0;
22     for(int i=0;i<n-1;i++){
23         min_index=i;
24         for(int j=i+1;j<n;j++){
25             if(arr[j]<arr[min_index]) min_index=j;
26         }
27         if(min_index!=i) swap(&arr[min_index],&arr[i]);
28     }
29 }
30
31 int main() {
32     int arr[] = {64, 34, 25, 12, 22, 11, 90};
33     int n = sizeof(arr)/sizeof(arr[0]);
34 //bubbleSort(arr, n);
35 SelectionSort(arr,n);
36 printf("\nSorted array: \n");
37 for (int i = 0; i < n; i++) {
38     printf("%d ", arr[i]);
39 }
40 printf("\n");
41 return 0;
42 }
```

Sorted array:

11 12 22 25 34 64 90

(base) fahee@Faheems-MacBook-Pro Programming_in_C %

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

The code demonstrates a simple implementation of the Selection Sort algorithm. It iteratively selects the smallest element from the unsorted portion of the array and places it at the beginning. This process continues until the entire array is sorted.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 12 | |
|--|---|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 12/04/2024 |
| Assignment Title (Full): | Write a C program to find the factorial of a number using recursion |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: <ul style="list-style-type: none">• Factorial Function (Fact):<ol style="list-style-type: none">1. Base case: If x is 0, return 1 (base case for factorial).2. Recursive case: Return x multiplied by Fact(x-1) to compute factorial recursively.• Main Function:<ol style="list-style-type: none">1. Prompt user to enter an integer x.2. Read the integer x using scanf.3. Call Fact(x) to compute the factorial of x.4. Print the computed factorial value with a formatted string using printf. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |

Output Response:

```
1 #include <stdio.h>
2
3 float Fact(int x){
4     if(x==0) return 1;
5     return x*Fact(x-1);
6 }
7
8 int main(){
9     int x;
10    printf("\nEnter number: ");
11    scanf("%d",&x);
12    printf("\nFactorial of %d is %f.\n",x,Fact(x));
13    return 0;
14 }
```

Enter number: 7

Factorial of 7 is 5040.000000.

) (base) fahee@Faheems-MacBook-Pro Programming_in_C %

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

This program calculates the factorial of a given non-negative integer x using a recursive approach. The factorial of x (denoted $x!$) is computed by multiplying x with the factorial of $x-1$, continuing until the base case of $0!$ is reached, which is defined as 1. Note that **float** is used for the return type of **Fact** to handle potentially large factorial values.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 13 | |
|--|--|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 15/04/2024 |
| Assignment Title (Full): | Write a C program using structures to print the pay slip of an employee after accepting details like id. no, name, designation, department and basic salary. |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: <ul style="list-style-type: none">Define two structs: Salary containing basic, gross, allowance, and tax; Employee containing id, name, designation, department, and a Salary struct.Create a Calculate function to compute allowance, tax, and gross salary based on the basic salary.Create an Insert function to input employee details including ID, name, designation, department, and basic salary, then calculate the salary components using Calculate.Create a Display function to show a tabular format of employee details including computed gross, basic, allowance, and tax.In main, prompt user for number of entries, create an array of Employee structures based on input, call Insert for each entry, then display all entries using Display. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |

Output Response:

```
1 #include <stdio.h>
2
3 struct Salary {
4     float basic;
5     float gross;
6     float allowance;
7     float tax;
8 };
9
10 struct Employee {
11     int id;
12     char name[50];
13     char designation[50];
14     char department[50];
15     struct Salary BS;
16 };
17
18 void Calculate(struct Employee *e) {
19     e->BS.allowance = (0.2) * e->BS.basic;
20     e->BS.tax = (0.1) * e->BS.basic;
21     e->BS.gross = e->BS.basic + e->BS.allowance - e->BS.tax;
22 }
23
24 void Insert(struct Employee *e) {
25     printf("\nEnter ID: ");
26     scanf("%d", &e->id);
27     getchar();
28     printf("\nEnter Name: ");
29     scanf("%[^n]s", e->name);
30     getchar();
31     printf("\nEnter Designation: ");
32     scanf("%[^n]s", e->designation);
33     getchar();
34     printf("\nEnter Department: ");
35     scanf("%[^n]s", e->department);
36     getchar();
37     printf("\nEnter Basic Salary: ");
38     scanf("%f", &e->BS.basic);
39     Calculate(e);
40 }
41
42 void Display(struct Employee e[], int size) {
43     printf("\n")
44     -----
45     ID      Name      Designation      Gross      Basic      Allowance      Tax\n\
46     -----
47
48     for(int i=0;i<size;i++){
49         printf(" %4d    %-10s    %-15s    %-10.2f    %-10.2f    %-10.2f\n",
50             e[i].id,e[i].name,e[i].designation,e[i].BS.gross,e[i].BS.basic,e[i].BS.allowance,e[i].BS.tax);
51     }
52     printf("\n");
53 }
54
55 int main() {
56     int n;
57     printf("\nEnter the number of entries: ");
58     scanf("%d",&n);
59     struct Employee emp[n];
60     for(int i=0;i<n;i++){
61         printf("\nEnter %d:\n",i+1);
62         Insert(&emp[i]);
63     }
64     Display(emp,n);
65     return 0;
66 }
```

```
Enter the number of entries: 3
```

```
Entry 1:
```

```
Enter ID: 10
```

```
Enter Name: Faheem
```

```
Enter Designation: CEO
```

```
Enter Department: Executive
```

```
Enter Basic Salary: 99999
```

```
Entry 2:
```

```
Enter ID: 20
```

```
Enter Name: Sanidhya
```

```
Enter Designation: COO
```

```
Enter Department: Executive
```

```
Enter Basic Salary: 90000
```

```
Entry 3:
```

```
Enter ID: 30
```

```
Enter Name: Rut
```

```
Enter Designation: Sales Head
```

```
Enter Department: Marketing
```

```
Enter Basic Salary: 75000
```

| ID | Name | Designation | Gross | Basic | Allowance | Tax |
|----|----------|-------------|-----------|----------|-----------|---------|
| 10 | Faheem | CEO | 109998.90 | 99999.00 | 19999.80 | 9999.90 |
| 20 | Sanidhya | COO | 99000.00 | 90000.00 | 18000.00 | 9000.00 |
| 30 | Rut | Sales Head | 82500.00 | 75000.00 | 15000.00 | 7500.00 |

```
○ (base) fahee@Faheems-MacBook-Pro:Programming_in_C % █
```

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

This program defines structures for Salary and Employee, calculates salary components using a separate function, and then allows the user to input multiple employee records with associated details and basic salary. The calculated salary details are displayed in a formatted table. This approach promotes modularity and encapsulation by organizing functionalities into discrete functions (Calculate, Insert, Display) and leveraging the power of structures to represent complex data.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma

Prof. Mahesh Arse

Prof. Sachin R. Gaikwad

Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 14 | |
|---|---|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 19/04/2024 |
| Assignment Title (Full): | 1. Write a program to demonstrate compile time memory allocation 2. Write a program to demonstrate runtime memory allocation |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: | |
| Compile Time Memory Allocation: | |
| 1. Define a function Display to print elements of an integer array. 2. In main, declare an integer array A of size 7. 3. Display the array before user input using Display(A, 7). 4. Prompt user to enter elements for A using scanf. 5. Display the modified array using Display(A, 7). | |
| Runtime Memory Allocation: | |
| 1. Define Display and Insert functions for printing and reading array elements respectively. 2. Declare variables n1, n2, *p1, *p2 for array size and pointers. 3. Prompt user to enter the size (n1) of the array. 4. Allocate memory for p1 using malloc and display its contents. 5. Input elements for p1 using Insert. 6. Allocate memory for p2 using calloc with the same size as p1. 7. Display and input elements for p2 as done for p1. 8. Prompt user to enter a new size (n2) for p1. 9. Reallocate memory for p1 using realloc based on n2. 10. Display and input elements for the reallocated p1. 11. Free allocated memory for p1 and p2 using free. | |

Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.)

Output Response:

1. Write a program to demonstrate compile time memory allocation

```
1 #include <stdio.h>
2
3 void Display(int arr[],int n){
4     printf("\nElements of the array are:\n");
5     for(int i=0;i<n;i++){
6         printf("%d ",arr[i]);
7     }
8     printf("\n");
9 }
10
11 int main(){
12     int A[7];
13     printf("\nDefault Initialization:\n");
14     Display(A,7);
15     printf("\nEnter the elements of the array:\n");
16     for(int i=0;i<7;i++){
17         scanf("%d",&A[i]);
18     }
19     Display(A,7);
20     return 0;
21 }
```

Default Initialization:

Elements of the array are:
1 1868672816 1 -2078766408 170754049 8289868 1

Enter the elements of the array:
1 2 3 4 5 6 7

Elements of the array are:
1 2 3 4 5 6 7

○ (base) fahee@Faheems-MacBook-Pro Programming_in_C %

2. Write a program to demonstrate runtime memory allocation:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void Display(int arr[],int n){
5     printf("\nElements of the array are:\n");
6     for(int i=0;i<n;i++){
7         printf("%d ",arr[i]);
8     }
9     printf("\n");
10 }
11
12 void Insert(int p[],int n){
13     printf("\nEnter the elements of the array:\n");
14     for(int i=0;i<n;i++){
15         scanf("%d",&p[i]);
16     }
17 }
18
19 int main(){
20     int n1,n2,*p1,*p2;
21     printf("\nPlease enter the size of the array: ");
22     scanf("%d",&n1);
23
24     p1=malloc(n1*sizeof(int));
25     printf("\n-----_MALLOC-----\n");
26     printf("\nDefault Initialization:\n");
27     Display(p1,n1);
28     Insert(p1,n1);
29     Display(p1,n1);
30
31     p2=calloc(n1,n1*sizeof(int));
32     printf("\n-----_CALLOC-----\n");
33     printf("\nDefault Initialization:\n");
34     Display(p2,n1);
35     Insert(p2,n1);
36     Display(p2,n1);
37
38     printf("\nEnter the modified size of the array: ");
39     scanf("%d",&n2);
40     p1=realloc(p1,n2*sizeof(int));
41     printf("\n-----_REALLOC-----\n");
42     printf("\nDefault Initialization:\n");
43     Display(p1,n2);
44     Insert(p1,n2);
45     Display(p1,n2);
46
47     free(p1);
48     free(p2);
49     return 0;
50 }
```

```
Please enter the size of the array: 5
```

```
-----MALLOC-----
```

```
Default Initialization:
```

```
Elements of the array are:
```

```
0 0 0 0 0
```

```
Enter the elements of the array:
```

```
10 20 30 40 50
```

```
Elements of the array are:
```

```
10 20 30 40 50
```

```
-----CALLOC-----
```

```
Default Initialization:
```

```
Elements of the array are:
```

```
0 0 0 0 0
```

```
Enter the elements of the array:
```

```
1 2 3 4 5
```

```
Elements of the array are:
```

```
1 2 3 4 5
```

```
Enter the modified size of the array: 7
```

```
-----REALLOC-----
```

```
Default Initialization:
```

```
Elements of the array are:
```

```
10 20 30 40 50 0 0
```

```
Enter the elements of the array:
```

```
10 20 30 40 50 60 70
```

```
Elements of the array are:
```

```
10 20 30 40 50 60 70
```

```
> (base) faheem@Faheems-MacBook-Pro Programming_in_C %
```

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

The first program demonstrates compile-time memory allocation using a statically declared array, while the second program demonstrates runtime memory allocation using dynamic memory allocation functions (malloc, calloc, realloc) to allocate memory during program execution. Dynamic allocation allows flexibility in managing memory based on runtime conditions, improving efficiency and adaptability in memory usage. However, it requires explicit memory management to avoid memory leaks.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte



SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

| Assignment No.: 15 | |
|---|---|
| Course Name | Programming in C Lab |
| Name of Student | Faheemuddin Sayyed |
| PRN No. | 23070122196 |
| Branch | CSE |
| Class | C-1 |
| Academic Year & Semester | 2023-2024 & Semester 2 |
| Date of Performance | 19/02/2024 |
| Assignment Title (Full): | Write a C Program to demonstrate Structure pointer. |
| Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithim of the C program in the Journal Notebook, and add its screenshot in the below theory response.) | |
| Theory Response: <ul style="list-style-type: none">Define two structures: Date for date components and Book containing book details and dates.Implement Fine function to calculate overdue fine based on return and issue dates.Implement Insert function to input book details and issue/return dates, then calculate fine using Fine function.Implement Display function to print book details and fine.In main, initialize books (b1, b2, b3) with predefined values and calculate fine using Fine.Use Insert to input details for b4, calculate fine, and display all book details. | |
| Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.) | |

Output Response:

```
1 #include <stdio.h>
2
3 struct Date {
4     int dd;
5     int mm;
6     int yy;
7 };
8
9 struct Book {
10    char name[100];
11    char auth_name[100];
12    struct Date issue;
13    struct Date rtrn;
14    float fine;
15 };
16
17 // Assuming return date to be 1 month after issue date
18 void Fine(struct Book *b) {
19     int days=0;
20     b->fine = 0;
21     if(b->rtrn.mm-b->issue.mm>1){
22         if(b->rtrn.dd>b->issue.dd){
23             days=(b->rtrn.mm-b->issue.mm-1)*30+b->rtrn.dd-b->issue.dd;
24         }
25         else{
26             days=(b->rtrn.mm-b->issue.mm-2)*30+(30-b->issue.dd+b->rtrn.dd);
27         }
28     }
29     else if(b->rtrn.mm-b->issue.mm==1){
30         if(b->rtrn.dd>b->issue.dd){
31             days=b->rtrn.dd-b->issue.dd;
32         }
33         else return;
34     }
35     else return;
36     b->fine=days*5; // 5rs per day fine
37 }
38
39 void Insert(struct Book *b){
40     printf("\nPlease enter Book name: ");
41     scanf("%[^\\n]s",b->name);
42     getchar(); // Consuming newline character
43     printf("\nPlease enter Author name: ");
44     scanf("%[^\\n]s",b->auth_name);
45     getchar();
46     printf("\nPlease enter Issuing date (dd mm yy): ");
47     scanf("%d %d %d",&b->issue.dd,&b->issue.mm,&b->issue.yy);
48     printf("\nPlease enter Return date (dd mm yy): ");
49     scanf("%d %d %d",&b->rtrn.dd,&b->rtrn.mm,&b->rtrn.yy);
50     Fine(b);
51 }
52
53 void Display(struct Book b){
54     printf("\nBook Name: %s", b.name);
55     printf("\nAuthor Name: %s", b.auth_name);
56     printf("\nIssuing Date: %d/%d/%d", b.issue.dd, b.issue.mm, b.issue.yy);
57     printf("\nReturn Date: %d/%d/%d", b.rtrn.dd, b.rtrn.mm, b.rtrn.yy);
58     printf("\nFine: \u20B9%.2f\n", b.fine);
59 }
60
61 int main() {
62     struct Book b1={"The Great Gatsby","F. Scott Fitzgerald", {1,1,22}, {15,1,22}, 0};
63     struct Book b2={"To Kill a Mockingbird","Harper Lee", {10,2,22}, {25,3,22}, 0};
64     struct Book b3={"1984","George Orwell", {20,3,22}, {10,5,22}, 0};
65     struct Book b4;
66
67     Fine(&b1);
68     Fine(&b2);
69     Fine(&b3);
70
71     Display(b1);
72     Display(b2);
73     Display(b3);
74
75     Insert(&b4);
76     Display(b4);
77     printf("\n");
78     return 0;
79 }
```

Book Name: The Great Gatsby
Author Name: F. Scott Fitzgerald
Issuing Date: 1/1/22
Return Date: 15/1/22
Fine: ₹0.00

Book Name: To Kill a Mockingbird
Author Name: Harper Lee
Issuing Date: 10/2/22
Return Date: 25/3/22
Fine: ₹75.00

Book Name: 1984
Author Name: George Orwell
Issuing Date: 20/3/22
Return Date: 10/5/22
Fine: ₹100.00

Please enter Book name: Don Quixote

Please enter Author name: Miguel de Cervantes

Please enter Issuing date (dd mm yy): 12 04 24

Please enter Return date (dd mm yy): 15 05 24

Book Name: Don Quixote
Author Name: Miguel de Cervantes
Issuing Date: 12/4/24
Return Date: 15/5/24
Fine: ₹15.00

○ (base) fahee@Faheems-MacBook-Pro Programming_in_C %

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

This program uses structure pointers to manage book records. It demonstrates functions to calculate fines based on issue and return dates, alongside functions to input and display book details. By organizing data into structures and using pointers, the program efficiently handles book records and associated operations.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte