

Name - Anjali Karwasi

Assignment No. 2

1

Registration Number - 23652008

Signature - Anjali

Course Code - OBC102

Course Title - Foundations of Computer Programming

PART-A

Q1. If an array is declared as `int arr[4][4]`, how many elements can be stored in it?

Ans C. 16

Q2. Consider an integer array, `arr[]`; the i th element can be accessed by writing

Ans C. `arr[i]`

Q3. It is possible to omit the mention of the _____ of the array at the time of initialization.

Ans C. Size

Q4. We can add a new data element to an already existing array, this is called _____.

Ans A. Inserting

Q5. Removing a data element from an already existing array, this is called _____.

Ans B. Deleting

Name - Anjali Kanwasai
Registration no. - 23652008

Anjali

Q6. _____ means finding an element in an array, whether the present or not.

Ans C. Searching

Q7. The _____ search is best for unsorted array.

Ans B. Linear

Q8. _____ is the process of arranging data in a specific order.

Ans D. Sorting

Q9. The function used to print a line of text on the output device is _____.

Ans G. Both (A) & (B)

Q10. The subscripts of the string start with the index _____.

Ans A. 0

Q11. Termination of a string is indicated by _____.

Ans A. \0

Name - Anjali Kanwasli
Registration no. - 23652008

(3)

Anjali

Q12. When two strings are compared, the comparison is based on _____.

Ans C. the length of the strings

Q13. To access union member variable, we generally use _____.

Ans A. Dot operator

Q14. _____ cannot be a structure member.

Ans B. function

Q15. _____ are types of data allowed inside a structure.

Ans D. all of these

Q16. If you pass a structure variable to a function, we actually pass _____.

Ans C. copy of structure variable

Q17. A do-while loop is useful when we want that the statements within the loop must be executed _____.

Ans b) At least once

Name - Anjali Kanwası

Registration no. - 23652008

(4)

Q18. In C programming, character input / output functions are _____ and _____.

Ans A. fgetc() and fputc()

Q19. In C programming, string input / output functions are _____ and _____.

Ans B. fgets() and fputs()

Q20. In C programming, formatted input / output functions are _____ and _____.

Ans C. fprintf() and fscanf()

PART - B

Q1. Write a well-defined recursive algorithm to compute Fibonacci series.

Ans 1

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

```
// Function to compute Fibonacci series recursively  
int fibonacci(int n) {
```

```
    // Base cases
```

```
    if (n <= 1)
```

```
        return n;
```

Name - Anjali Kanwasi
Registration no. - 23652008

(5)

Anjali

```
// Recursive step  
return fibonacci(n - 1) + fibonacci(n - 2);  
}
```

```
// Function to print Fibonacci series upto n terms  
void printFibonacciSeries(int n) {  
    printf("Fibonacci Series upto %d terms: ", n);  
    for (int i = 0; i < n; i++) {  
        printf("%d", fibonacci(i))  
    }  
    printf("\n");  
}
```

// Main function to test the recursive Fibonacci algorithm

```
int main() {  
    int n;
```

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

// Check for valid input
if (n < 0) {

Name - Anjali Kanuasi
Registration no. - 23652008

⑥

```
    printf("Invalid input, Please enter a  
    ↗ non-negative integer.\n");  
}  
else {  
    // Print Fibonacci series  
    printf(fibonacci Series(n));  
}  
  
return 0;  
}
```

Q3. Differentiate between structure and union definition.

Ans3 Both structures and unions are used to group different data types together.

Structure Definition :

→ Syntax :

```
struct structure-name {  
    datatype member1;
```

Name - Anjali Kanwası ①

Registration no. - 23652008

Anjali

```
datatype member2;  
// ... additional members  
};
```

→ Memory Allocation :

1. Each member of a structure is assigned a unique memory location.
2. The size of a structure is the sum of the sizes of its individual members plus any padding added for alignment.

→ Usage :

Suitable when you need to store and access different types of data for each member independently.

→ Example :

```
struct Point {  
    int x;  
    int y;  
};
```

Name - Anjali Kanwari ⑧

Registration no. - 23652008

Anjali

Union Definition:

→ Syntax:

```
union union_name {  
    datatype member 1;  
    datatype member 2;  
    // ... additional members  
};
```

→ Memory Allocation:

1. All members of a union share the same memory location.
2. The size of a union is determined by the size of its largest member.

→ Usage:

Suitable when you want to share store different types of data, but only one member at a time is relevant.

→ Example:

```
union Status {  
    int errorCode;  
    char message[50];  
};
```

Name - Anyali Kanwasri

Registration no. - 23652008

(9)

Anyali

Key Differences :

① Memory Allocation :

In structures, each member has its own memory space.

In unions, all members share the same memory space.

② Size :

Structure size is the sum of sizes of all its members plus padding for alignment.

Union size is determined by the size of its largest member.

③ Usage :

Structures are used when you want to store and access different types of data independently.

Unions are used when you want to save memory and only one member is relevant at a time.

④ Access :

In structures, you can access any member at any time. In Unions, only one member can be accessed at a time.

Name - Anyali Kanwasji
Registration no. - 23652008

(10)

Q5.

Write a C program to find the length of a given string without using string functions.

Ans 5

```
#include <stdio.h>

int findStringLength(char *str) {
    int length = 0;
    while (str[length] != '\0') {
        length++;
    }
    return length;
}

int main() {
    char str[100];
    printf("Enter the string: ");
    gets(str, sizeof(str), stdin);
    int length = findStringLength(str);
    printf("Length of the string: %d\n", length);
    return 0;
}
```

Name - Anjali Kanwasai
 Registration no. - 23652008

Anjali

Q6.

Write a program to perform matrix addition.

Ans6

```
#include <stdio.h>

void inputMatrix(int rows, int cols, int matrix[][cols]) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("Enter element of position ");
            printf "[%d][%d]: ", i, j);
            scanf ("%d", &matrix[i][j]);
        }
    }
}

void addMatrices(
    int rows,
    int cols,
    int matrix1[][cols],
    int matrix2[][cols],
    int result[][cols],
) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
}
```

(12)

Name - Anjali Kanwasi
 Registration no. - 23652008

Anjali

}

}

}

```

int main() {
    int rows, cols;
    printf("Enter the number of rows and columns: ");
    scanf("%d%d", &rows, &cols);

    int matrix1[rows][cols], matrix2[rows][cols],
        result[rows][cols];

    printf("Enter the elements of first matrix:\n");
    inputMatrix(rows, cols, matrix1);

    printf("Enter the elements of second matrix:\n");
    inputMatrix(rows, cols, matrix2);

    addMatrices(rows, cols, matrix1, matrix2, result);

    printf("Resultant Matrix :\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }
}

```

Name - Anjali Kanwasí

Registration no. - 23652008

(13)

Anjali

}

return 0;

}

Q7. Write a program to perform matrix multiplication.

Ans

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

```
void inputMatrix(int rows, int cols, int matrix [][cols]) {  
    for(int i = 0; i < rows; i++) {
```

```
        for(int j = 0; j < cols; j++) {
```

printf ("Enter element at position")

[%d][%d]: ", i, j);

```
scanf ("%d", &matrix[i][j]);
```

}

}

```
}
```

```
void multiplyMatrices(
```

```
    int rows1,
```

```
    int cols1,
```

```
    int matrix1 [][cols1],
```

```
    int rows2,
```

```
    int cols2,
```

Name - Anjali Kanwasli
Registration no. - 23652008

(14)

```
int matrix1[ ][cols1],  
int result[ ][cols2],  
) {  
    for (int i = 0; i < rows1; i++) {  
        for (int j = 0; j < cols2; j++) {  
            result[i][j] = 0;  
            for (int k = 0; k < cols1; k++) {  
                result[i][j] += matrix1[i][k]  
                * matrix2[k][j];  
            }  
        }  
    }  
}  
  
int main() {  
    int rows1, cols1, row2, cols2;  
    printf(  
        "Enter the number of rows and columns  
        for first matrix : "  
    );  
    scanf("%d%d", &rows1, &cols1);  
    printf(  
        "Enter the number of rows and  
        columns for second matrix : "  
    );  
}
```

Name - Anjali Karuasi

(15)

Registration no. - 23652008

Anjali

```
 );
scanf ("%d%d", &rows2, &cols2);

int matrix1[rows1][cols1], matrix2[rows2][cols2],
      result[rows1][cols2];

printf ("Enter the elements of first matrix :\n");
inputMatrix (rows1, cols1, matrix1);

printf ("Enter the elements of second matrix :\n");
inputMatrix (rows2, cols2, matrix2);

multiplyMatrices (
    rows1, cols1, matrix1, rows2, cols2, matrix2, result
);

printf ("Resultant Matrix :\n");
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        printf ("%d\t", matrix[i][j]);
    }
    printf ("\n");
}

return 0;
```