**Question Paper 1 - String Operations**

**Question:**

1. **Write a program to store N unique string values in a data structure of your choice** (5 Marks).  
   Perform the following operations on the constructed data structure:
   1. Find the string that has the maximum length (5 Marks).
   2. Delete string’s that starts with letter 'A' (5 Marks).
   3. Split the data into two parts at a given position (5 Marks).
   4. Deallocate all the memory allocated dynamically (5 Marks).

**Header File: string\_ops.h**

c

Copy code

#ifndef STRING\_OPS\_H

#define STRING\_OPS\_H

// Function declarations

void store\_strings(char \*\*\*arr, int n);

void find\_max\_length\_string(char \*\*arr, int n);

void delete\_strings\_starting\_with\_A(char \*\*arr, int \*n);

void split\_strings\_at\_position(char \*\*arr, int n, int pos);

void deallocate\_memory(char \*\*arr, int n);

#endif

**Implementation File: string\_ops.c**

c

Copy code

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "string\_ops.h"

// Function to store N unique strings

void store\_strings(char \*\*\*arr, int n) {

\*arr = (char \*\*)malloc(n \* sizeof(char \*));

for (int i = 0; i < n; i++) {

(\*arr)[i] = (char \*)malloc(100 \* sizeof(char));

printf("Enter string %d: ", i + 1);

scanf("%s", (\*arr)[i]);

}

}

// Function to find the string with the maximum length

void find\_max\_length\_string(char \*\*arr, int n) {

int max\_len = 0, index = -1;

for (int i = 0; i < n; i++) {

int len = strlen(arr[i]);

if (len > max\_len) {

max\_len = len;

index = i;

}

}

if (index != -1)

printf("The string with maximum length is: %s\n", arr[index]);

}

// Function to delete strings starting with 'A'

void delete\_strings\_starting\_with\_A(char \*\*arr, int \*n) {

int count = 0;

for (int i = 0; i < \*n; i++) {

if (arr[i][0] == 'A' || arr[i][0] == 'a') {

free(arr[i]);

for (int j = i; j < \*n - 1; j++) {

arr[j] = arr[j + 1];

}

(\*n)--;

i--; // Decrement to recheck the shifted string

}

}

}

// Function to split the data at a given position

void split\_strings\_at\_position(char \*\*arr, int n, int pos) {

if (pos < 0 || pos >= n) {

printf("Invalid position.\n");

return;

}

printf("First part:\n");

for (int i = 0; i < pos; i++) {

printf("%s\n", arr[i]);

}

printf("Second part:\n");

for (int i = pos; i < n; i++) {

printf("%s\n", arr[i]);

}

}

// Function to deallocate memory

void deallocate\_memory(char \*\*arr, int n) {

for (int i = 0; i < n; i++) {

free(arr[i]);

}

free(arr);

}

**Main File: main.c**

c

Copy code

#include <stdio.h>

#include <stdlib.h>

#include "string\_ops.h"

int main() {

int n;

printf("Enter the number of strings: ");

scanf("%d", &n);

char \*\*arr = NULL;

store\_strings(&arr, n);

find\_max\_length\_string(arr, n);

delete\_strings\_starting\_with\_A(arr, &n);

int pos;

printf("Enter position to split the array: ");

scanf("%d", &pos);

split\_strings\_at\_position(arr, n, pos);

deallocate\_memory(arr, n);

return 0;

}

**Question Paper 2 - Integer Operations**

**Question:**

1. **Write a program to store N integer values in a data structure of your choice** (5 Marks).  
   Perform the following operations on the constructed data structure:
   1. Delete all the occurrences of a digit (5 Marks).
   2. Find two integers (X, Y) that sum to M (5 Marks).
   3. Count number of even and odd numbers (5 Marks).
   4. Deallocate all the memory allocated dynamically (5 Marks).

**Header File: integer\_ops.h**

c

Copy code

#ifndef INTEGER\_OPS\_H

#define INTEGER\_OPS\_H

// Function declarations

void store\_integers(int \*\*arr, int \*n);

void delete\_digit\_occurrences(int \*\*arr, int \*n, int digit);

void find\_pair\_with\_sum(int \*arr, int n, int m);

void count\_even\_odd(int \*arr, int n);

void deallocate\_memory(int \*arr);

#endif

**Implementation File: integer\_ops.c**

c

Copy code

#include <stdio.h>

#include <stdlib.h>

#include "integer\_ops.h"

// Function to store N integers

void store\_integers(int \*\*arr, int \*n) {

printf("Enter the number of integers: ");

scanf("%d", n);

\*arr = (int \*)malloc(\*n \* sizeof(int));

printf("Enter the integers:\n");

for (int i = 0; i < \*n; i++) {

scanf("%d", &(\*arr)[i]);

}

}

// Function to delete all occurrences of a digit

void delete\_digit\_occurrences(int \*\*arr, int \*n, int digit) {

int count = 0;

for (int i = 0; i < \*n; i++) {

if ((\*arr)[i] == digit) {

for (int j = i; j < \*n - 1; j++) {

(\*arr)[j] = (\*arr)[j + 1];

}

(\*n)--;

i--;

}

}

}

// Function to find two integers (X, Y) that sum to M

void find\_pair\_with\_sum(int \*arr, int n, int m) {

for (int i = 0; i < n; i++) {

for (int j = i + 1; j < n; j++) {

if (arr[i] + arr[j] == m) {

printf("Pair found: (%d, %d)\n", arr[i], arr[j]);

return;

}

}

}

printf("No pair found with sum %d.\n", m);

}

// Function to count even and odd numbers

void count\_even\_odd(int \*arr, int n) {

int even\_count = 0, odd\_count = 0;

for (int i = 0; i < n; i++) {

if (arr[i] % 2 == 0) {

even\_count++;

} else {

odd\_count++;

}

}

printf("Even numbers: %d, Odd numbers: %d\n", even\_count, odd\_count);

}

// Function to deallocate memory

void deallocate\_memory(int \*arr) {

free(arr);

}

**Main File: main.c**

c

Copy code

#include <stdio.h>

#include <stdlib.h>

#include "integer\_ops.h"

int main() {

int \*arr = NULL;

int n;

store\_integers(&arr, &n);

int digit;

printf("Enter a digit to delete its occurrences: ");

scanf("%d", &digit);

delete\_digit\_occurrences(&arr, &n, digit);

int m;

printf("Enter the sum value M: ");

scanf("%d", &m);

find\_pair\_with\_sum(arr, n, m);

count\_even\_odd(arr, n);

deallocate\_memory(arr);

return 0;

}

**Compilation Instructions:**

1. Save the following files:
   * string\_ops.h, string\_ops.c, main.c for the string operations questions.
   * integer\_ops.h, integer\_ops.c, main.c for the integer operations questions.
2. Use the following commands to compile:

bash

Copy code

gcc main.c string\_ops.c -o string\_program

./string\_program

gcc main.c integer\_ops.c -o integer\_program

./integer\_program