**Assignment 5 : Dynamic Memory Allocation And File Management:**

1. Assign Value to the particular place using malloc.

#include <stdlib.h>

int main(){

int \*ptr;

ptr = malloc(15 \* sizeof(\*ptr)); /\* a block of 15 integers \*/

if (ptr != NULL) {

\*(ptr + 5) = 480; /\* assign 480 to sixth integer \*/

printf("Value of the 6th integer is %d",\*(ptr + 5));

}

}

1. Calculates the sum of an arithmetic sequence using calloc.

#include <stdio.h>

int main() {

int i, \* ptr, sum = 0;

ptr = calloc(10, sizeof(int));

if (ptr == NULL) {

printf("Error! memory not allocated.");

exit(0);

}

printf("Building and calculating the sequence sum of the first 10 terms \ n ");

for (i = 0; i < 10; ++i) { \* (ptr + i) = i;

sum += \* (ptr + i);

}

printf("Sum = %d", sum);

free(ptr);

return 0;

}

1. Write a C program to resize the memory block.

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*ptr = (int\*) malloc(3 \* sizeof(int));

ptr[0] = 1;

ptr[1] = 2;

ptr[2] = 3;

// resize the memory block to hold 5 integers

ptr = (int\*) realloc(ptr, 5 \* sizeof(int));

ptr[3] = 4;

ptr[4] = 5;

for (int i = 0; i< 5; i++) {

printf("%d ", ptr[i]);

}

// free the memory block

free(ptr);

return 0;

}

File Management:

1. C Program to illustrate file opening

#include <stdio.h>

#include <stdlib.h>

int main()

{

// file pointer variable to store the value returned by

// fopen

FILE\* fptr;

// opening the file in read mode

fptr = fopen("C:\filename.txt", "r");

// checking if the file is opened successfully

if (fptr == NULL) {

printf("The file is not opened. The program will "

"now exit.");

exit(0);

}

return 0;

}

1. // C Program to create a file

#include <stdio.h>

#include <stdlib.h>

int main()

{

// file pointer

FILE\* fptr;

// creating file using fopen() access mode "w"

fptr = fopen("C:\file.txt", "w");

// checking if the file is created

if (fptr == NULL) {

printf("The file is not opened. The program will " "exit now");

exit(0);

}

else {

printf("The file is created Successfully.");

}

return 0;

}

1. File Implementation:

#include <stdio.h>

int main() {

FILE \*file;

char data[100];

// Opening a file for writing

file = fopen("c:\example.txt", "w");

if (file == NULL) {

printf("Error opening file for writing.\n");

return 1; // Exit program with an error code

}

// Inserting data into the file

fprintf(file, "Hello, this is some data written to the file.\n");

fprintf(file, "12345\n");

fprintf(file, "3.14\n");

// Closing the file after writing

fclose(file);

// Opening the file for reading

file = fopen("example.txt", "r");

if (file == NULL) {

printf("Error opening file for reading.\n");

return 1; // Exit program with an error code

}

// Extracting data from the file

while (fgets(data, sizeof(data), file) != NULL) {

printf("Data from file: %s", data);

}

// Closing the file after reading

fclose(file);

return 0; // Exit program successfully

}