**Assigment-1 Date: 19/01/2024**

**Pratical-1:**

**Aim:Assign Value to the particular place using malloc.**

**Code:**

**#include <stdlib.h>**

**int main(){**

**int \*ptr;**

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

**if (ptr != NULL) {**

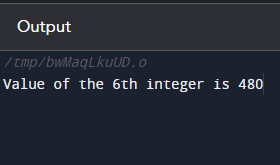
**\*(ptr + 5) = 480; /\* assign 480 to sixth integer \*/**

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

**}**

**}**

**Output:**



**Pratical-2:**

**Aim:Calculates the sum of an arithmetic sequence using calloc.**

**Code:**

**#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);**

**}**

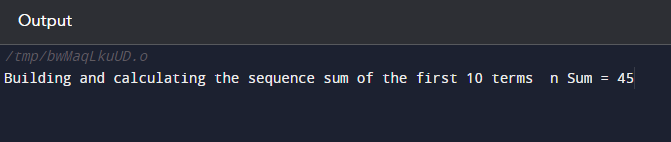
**printf("Sum = %d", sum);**

**free(ptr);**

**return 0;**

**}**

**Output:**



**Pratical-3:**

**Aim:Write a C program to resize the memory block.**

**Code:**

**#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]);**

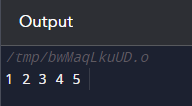
**}**

**// free the memory block**

**free(ptr);**

**return 0;**

**}**

**Output:**

**File Management:**

**Pratical-4:**

**Aim:C Program to illustrate file opening**

**Code:// 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("filename.txt", "r");**

**// checking if the file is opened successfully**

**if (fptr == NULL) {**

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

**"now exit.");**

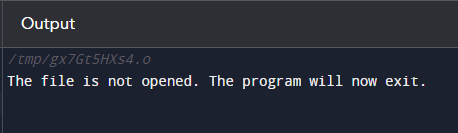
**exit(0);**

**}**

**return 0;**

**}**

**Output:**



**Pratical-5:**

**// 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("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 {**

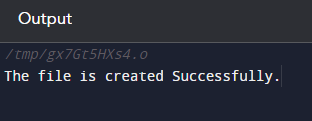
**printf("The file is created Successfully.");**

**}**

**return 0;**

**}**

**Output:**



**Pratical-6:**

**Aim: File Implementation:**

**Code:**

**#include &lt;stdio.h&gt;**

**int main() {**

**FILE \*file;**

**char data[100];**

**// Opening a file for writing**

**file = fopen(&quot;c:\example.txt&quot;, &quot;w&quot;);**

**if (file == NULL) {**

**printf(&quot;Error opening file for writing.\n&quot;);**

**return 1; // Exit program with an error code**

**}**

**// Inserting data into the file**

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

**fprintf(file, &quot;12345\n&quot;);**

**fprintf(file, &quot;3.14\n&quot;);**

**// Closing the file after writing**

**fclose(file);**

**// Opening the file for reading**

**file = fopen(&quot;example.txt&quot;, &quot;r&quot;);**

**if (file == NULL) {**

**printf(&quot;Error opening file for reading.\n&quot;);**

**return 1; // Exit program with an error code**

**}**

**// Extracting data from the file**

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

**printf(&quot;Data from file: %s&quot;, data);**

**}**

**// Closing the file after reading**

**fclose(file);**

**return 0; // Exit program successfully**

**}**

**Output:**

