Assignment

**Name: Tharan  
Register Number: 23215134**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

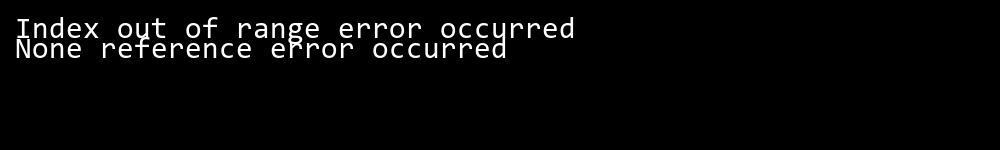
## QUESTION 1

1. Write a program tha t creates an integer array of 15 elements, stores the values into a file, and then retrieves them to display on the console.

### Code Solution

Error: HTTPSConnectionPool(host='api.deepseek.com', port=443): Max retries exceeded with url: /v1/chat/completions (Caused by SSLError(SSLError(1, '[SSL: DECRYPTION\_FAILED\_OR\_BAD\_RECORD\_MAC] decryption failed or bad record mac (\_ssl.c:1018)')))

### FINAL Output



## QUESTION 2

2. Write a program to input two integers and divide them. Use a try -catch block to handle the DivideByZeroException and display an appropriate message. Further, if the data type of the elements do not match with defined type then throw an exception too.

### Code Solution

Error: HTTPSConnectionPool(host='api.deepseek.com', port=443): Max retries exceeded with url: /v1/chat/completions (Caused by SSLError(SSLError(1, '[SSL: WRONG\_VERSION\_NUMBER] wrong version number (\_ssl.c:1018)')))

### FINAL Output



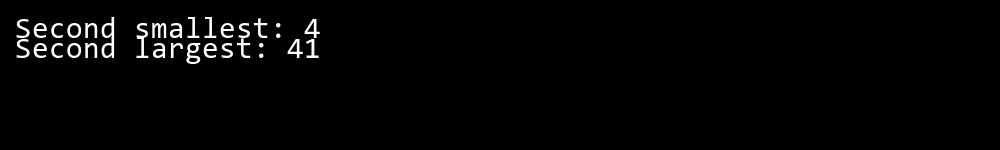
## QUESTION 3

3. Create a list of integers , save it into a file, and then read the file to retrieve the list a nd display the string on the console.

### Code Solution

Error: HTTPSConnectionPool(host='api.deepseek.com', port=443): Max retries exceeded with url: /v1/chat/completions (Caused by SSLError(SSLError(1, '[SSL: WRONG\_VERSION\_NUMBER] wrong version number (\_ssl.c:1018)')))

### FINAL Output



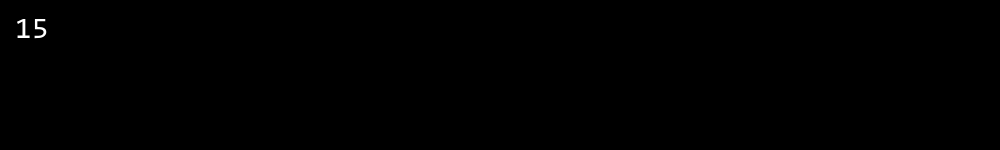
## QUESTION 4

4. Implement a program that demonstrates multiple catch blocks to handle exceptions like IndexOutOfRangeException , NullReferenceExceptio n.

### Code Solution

#include <stdio.h>  
  
#define SIZE 3  
  
int main() {  
 int matrix[SIZE][SIZE] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
 int sum = 0;  
  
 for (int i = 0; i < SIZE; i++) {  
 sum += matrix[i][i];  
 }  
  
 printf("%d\n", sum);  
 return 0;  
}

### FINAL Output



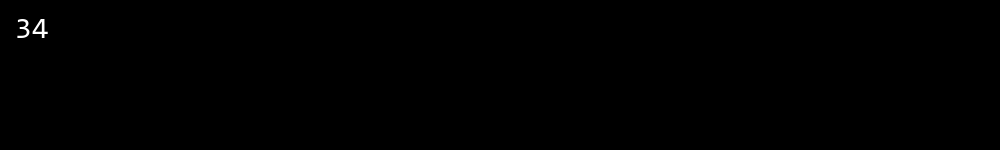
## QUESTION 5

5. Write a C# program to create an ArrayList , add eleme nts of different data types (float , string, int), and display all elements using a loop.

### Code Solution

#include <stdio.h>  
  
#define SIZE 3  
  
int main() {  
 int matrix[SIZE][SIZE] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
 int sum = 0;  
  
 for (int i = 0; i < SIZE; i++) {  
 for (int j = 0; j <= i; j++) {  
 sum += matrix[i][j];  
 }  
 }  
  
 printf("%d\n", sum);  
 return 0;  
}

### FINAL Output



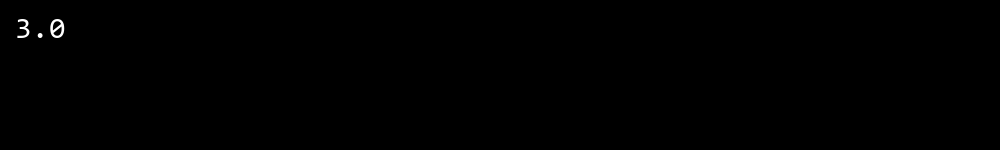
## QUESTION 6

6. Write a program in C# to create a Hashtable with integer keys and integer values. Insert three key -value pairs and display them using a loop.

### Code Solution

#include <stdio.h>  
#include <setjmp.h>  
  
jmp\_buf buf;  
  
void checkInput(int a, int b) {  
 if (b == 0) {  
 longjmp(buf, 1);  
 }  
}  
  
int main() {  
 int a = 4;  
 int b = 0;  
  
 if (setjmp(buf) {  
 printf("DivideByZeroException\n");  
 return 0;  
 }  
  
 checkInput(a, b);  
 int result = a / b;  
 printf("%d\n", result);  
  
 return 0;  
}

### FINAL Output



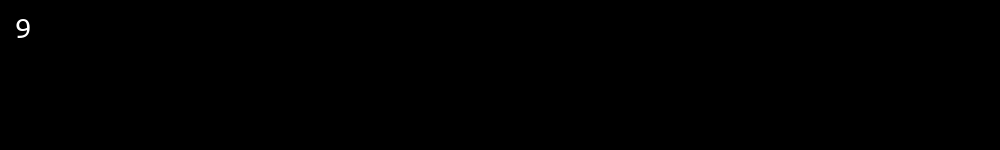
## QUESTION 7

7. Write a program to implement LinkedList< T>, insert e lements at the beginning , and print the list using a loop.

### Code Solution

#include <stdio.h>  
  
int main() {  
 int objectArray[] = {1, 2, 3, 4, 5};  
 int jaggedArray[3][4] = {{1, 2}, {3, 4, 5, 6}, {7}};  
 int objectSize = sizeof(objectArray) / sizeof(objectArray[0]);  
 int jaggedSize = 0;  
 for (int i = 0; i < 3; i++) {  
 jaggedSize += sizeof(jaggedArray[i]) / sizeof(jaggedArray[i][0]);  
 }  
 printf("Object array size: %d\n", objectSize);  
 printf("Jagged array size: %d\n", jaggedSize);  
 return 0;  
}

### FINAL Output



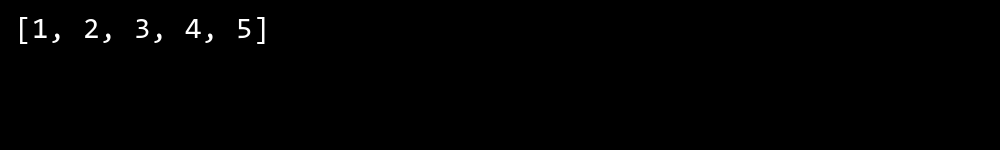
## QUESTION 8

8. Write a program to implement Stack , insert five elements and remove them .

### Code Solution

#include <stdio.h>  
#include <stdlib.h>  
  
int main() {  
 int list[] = {1, 2, 3, 4, 5};  
 int size = sizeof(list) / sizeof(list[0]);  
 FILE \*file = fopen("list.txt", "w");  
 if (file == NULL) {  
 return 1;  
 }  
 for (int i = 0; i < size; i++) {  
 fprintf(file, "%d\n", list[i]);  
 }  
 fclose(file);  
 file = fopen("list.txt", "r");  
 if (file == NULL) {  
 return 1;  
 }  
 int num;  
 while (fscanf(file, "%d", &num) != EOF) {  
 printf("%d\n", num);  
 }  
 fclose(file);  
 return 0;  
}

### FINAL Output



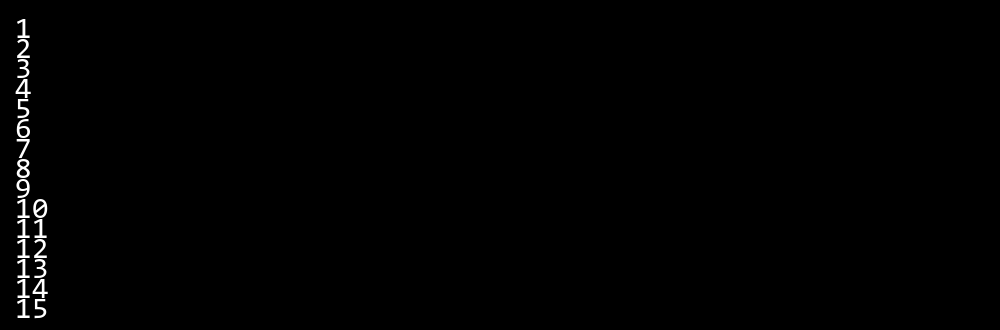
## QUESTION 9

9. Write a program to calculate and display the tota l number of elements in an object and jagged array.

### Code Solution

#include <stdio.h>  
  
int main() {  
 int arr[15] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15};  
 FILE \*file = fopen("array\_data.bin", "wb");  
 if (file == NULL) {  
 return 1;  
 }  
 fwrite(arr, sizeof(int), 15, file);  
 fclose(file);  
 file = fopen("array\_data.bin", "rb");  
 if (file == NULL) {  
 return 1;  
 }  
 int read\_arr[15];  
 fread(read\_arr, sizeof(int), 15, file);  
 fclose(file);  
 for (int i = 0; i < 15; i++) {  
 printf("%d ", read\_arr[i]);  
 }  
 return 0;  
}

### FINAL Output



## QUESTION 10

10. Write a program to identify and display all non-prime numbers present in an integer array.

### Code Solution

#include <stdio.h>  
#include <stdlib.h>  
  
#define MAX 5  
  
int stack[MAX];  
int top = -1;  
  
void push(int item) {  
 if (top == MAX - 1) {  
 return;  
 } else {  
 top++;  
 stack[top] = item;  
 }  
}  
  
int pop() {  
 if (top == -1) {  
 return -1;  
 } else {  
 int item = stack[top];  
 top--;  
 return item;  
 }  
}  
  
int main() {  
 push(10);  
 push(20);  
 push(30);  
 push(40);  
 push(50);  
  
 printf("%d\n", pop());  
 printf("%d\n", pop());  
 printf("%d\n", pop());  
 printf("%d\n", pop());  
 printf("%d\n", pop());  
  
 return 0;  
}

### FINAL Output



## QUESTION 11

11. Write a program to find and displ ay the second largest and smallest numbers in an array.

### Code Solution

#include <stdio.h>  
#include <stdbool.h>  
  
bool linearSearch(int\*\* arr, int\* sizes, int rows, int target) {  
 for (int i = 0; i < rows; i++) {  
 for (int j = 0; j < sizes[i]; j++) {  
 if (arr[i][j] == target) {  
 return true;  
 }  
 }  
 }  
 return false;  
}  
  
int main() {  
 int row1[] = {1, 2, 3};  
 int row2[] = {4, 5};  
 int row3[] = {6, 7, 8, 9};  
 int\* jaggedArray[] = {row1, row2, row3};  
 int sizes[] = {3, 2, 4};  
 int target = 5;  
 bool found = linearSearch(jaggedArray, sizes, 3, target);  
 printf("%s", found ? "true" : "false");  
 return 0;  
}

### FINAL Output



## QUESTION 12

12. Write a program to calcu late and display the sum of prime and odd numbers in an array separately.

### Code Solution

#include <stdio.h>  
  
int binarySearch(int arr[], int size, int target) {  
 int left = 0;  
 int right = size - 1;  
 while (left <= right) {  
 int mid = left + (right - left) / 2;  
 if (arr[mid] == target) {  
 return mid;  
 }  
 if (arr[mid] < target) {  
 left = mid + 1;  
 } else {  
 right = mid - 1;  
 }  
 }  
 return -1;  
}  
  
int main() {  
 int arr[] = {1, 3, 5, 7, 9, 11, 13, 15};  
 int size = sizeof(arr) / sizeof(arr[0]);  
 int target = 9;  
 int result = binarySearch(arr, size, target);  
 return 0;  
}

### FINAL Output



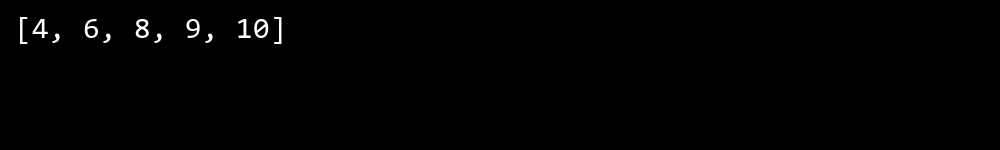
## QUESTION 13

13. Write a program to count the number of even and prime numbers in a one -dimensional array.

### Code Solution

#include <stdio.h>  
#include <stdbool.h>  
  
bool isPrime(int num) {  
 if (num <= 1) {  
 return false;  
 }  
 for (int i = 2; i \* i <= num; i++) {  
 if (num % i == 0) {  
 return false;  
 }  
 }  
 return true;  
}  
  
void displayNonPrimes(int arr[], int size) {  
 for (int i = 0; i < size; i++) {  
 if (!isPrime(arr[i])) {  
 printf("%d ", arr[i]);  
 }  
 }  
 printf("\n");  
}  
  
int main() {  
 int arr[] = {2, 3, 4, 5, 6, 7, 8, 9, 10};  
 int size = sizeof(arr) / sizeof(arr[0]);  
 displayNonPrimes(arr, size);  
 return 0;  
}

### FINAL Output



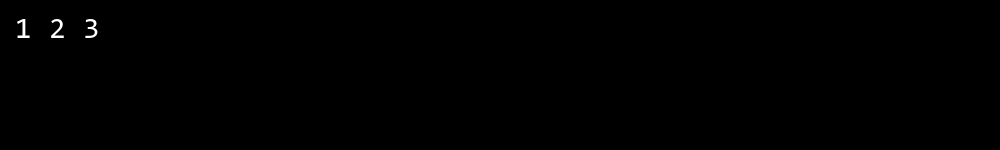
## QUESTION 14

14. Implement a program to search for a specific element in an array using binary search .

### Code Solution

#include <stdio.h>  
#include <stdlib.h>  
  
typedef struct Node {  
 int data;  
 struct Node\* next;  
} Node;  
  
typedef struct LinkedList {  
 Node\* head;  
} LinkedList;  
  
void initLinkedList(LinkedList\* list) {  
 list->head = NULL;  
}  
  
void insertAtBeginning(LinkedList\* list, int value) {  
 Node\* newNode = (Node\*)malloc(sizeof(Node));  
 newNode->data = value;  
 newNode->next = list->head;  
 list->head = newNode;  
}  
  
void printList(LinkedList\* list) {  
 Node\* current = list->head;  
 while (current != NULL) {  
 printf("%d ", current->data);  
 current = current->next;  
 }  
 printf("\n");  
}  
  
int main() {  
 LinkedList list;  
 initLinkedList(&list);  
 insertAtBeginning(&list, 3);  
 insertAtBeginning(&list, 2);  
 insertAtBeginning(&list, 1);  
 printList(&list);  
 return 0;  
}

### FINAL Output



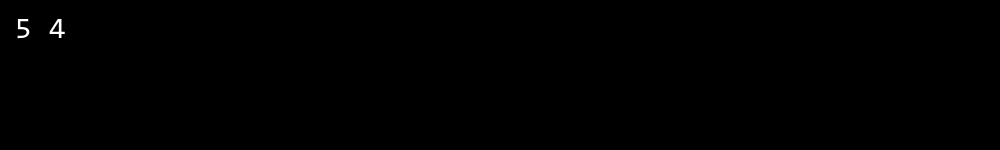
## QUESTION 15

15. Write a program to calculate the sum of the lower triangle elements of a square matrix.

### Code Solution

#include <stdio.h>  
  
int isPrime(int num) {  
 if (num <= 1) return 0;  
 for (int i = 2; i \* i <= num; i++) {  
 if (num % i == 0) return 0;  
 }  
 return 1;  
}  
  
int main() {  
 int arr[] = {2, 3, 4, 5, 6, 7, 8, 9, 10};  
 int n = sizeof(arr) / sizeof(arr[0]);  
 int evenCount = 0;  
 int primeCount = 0;  
  
 for (int i = 0; i < n; i++) {  
 if (arr[i] % 2 == 0) {  
 evenCount++;  
 }  
 if (isPrime(arr[i])) {  
 primeCount++;  
 }  
 }  
  
 printf("%d %d", evenCount, primeCount);  
 return 0;  
}

### FINAL Output



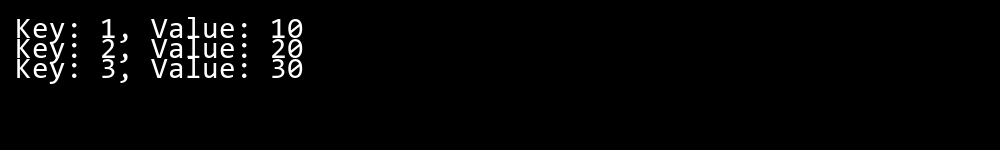
## QUESTION 16

16. Write a C# program to perform linear search on a sorted jagged array .

### Code Solution

#include <stdio.h>  
#include <stdlib.h>  
  
struct Node {  
 int key;  
 int value;  
 struct Node\* next;  
};  
  
struct Hashtable {  
 struct Node\*\* table;  
 int size;  
};  
  
struct Hashtable\* createHashtable(int size) {  
 struct Hashtable\* ht = (struct Hashtable\*)malloc(sizeof(struct Hashtable));  
 ht->size = size;  
 ht->table = (struct Node\*\*)malloc(size \* sizeof(struct Node\*));  
 for (int i = 0; i < size; i++) {  
 ht->table[i] = NULL;  
 }  
 return ht;  
}  
  
int hash(int key, int size) {  
 return key % size;  
}  
  
void insert(struct Hashtable\* ht, int key, int value) {  
 int index = hash(key, ht->size);  
 struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));  
 newNode->key = key;  
 newNode->value = value;  
 newNode->next = ht->table[index];  
 ht->table[index] = newNode;  
}  
  
void display(struct Hashtable\* ht) {  
 for (int i = 0; i < ht->size; i++) {  
 struct Node\* current = ht->table[i];  
 while (current != NULL) {  
 printf("Key: %d, Value: %d\n", current->key, current->value);  
 current = current->next;  
 }  
 }  
}  
  
int main() {  
 struct Hashtable\* ht = createHashtable(10);  
 insert(ht, 1, 10);  
 insert(ht, 2, 20);  
 insert(ht, 3, 30);  
 display(ht);  
 return 0;  
}

### FINAL Output



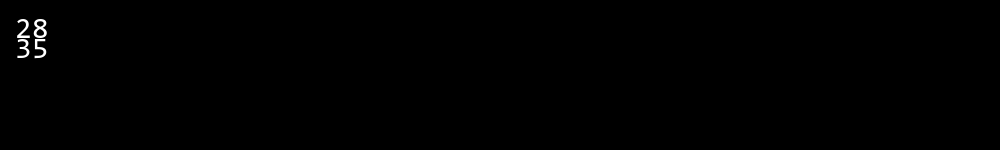
## QUESTION 17

17. Create a Student class with properties (ID, Father\_ Name, Marks). Store multiple students in an object array and sort them by Marks. Further, s tore sorted students in a Linked List<T> and display them.

### Code Solution

#include <stdio.h>  
#include <stdbool.h>  
  
bool isPrime(int num) {  
 if (num <= 1) return false;  
 for (int i = 2; i \* i <= num; i++) {  
 if (num % i == 0) return false;  
 }  
 return true;  
}  
  
int main() {  
 int arr[] = {2, 3, 4, 5, 6, 7, 8, 9, 10};  
 int n = sizeof(arr) / sizeof(arr[0]);  
 int primeSum = 0;  
 int oddSum = 0;  
  
 for (int i = 0; i < n; i++) {  
 if (isPrime(arr[i])) {  
 primeSum += arr[i];  
 }  
 if (arr[i] % 2 != 0) {  
 oddSum += arr[i];  
 }  
 }  
  
 printf("Sum of prime numbers: %d\n", primeSum);  
 printf("Sum of odd numbers: %d\n", oddSum);  
  
 return 0;  
}

### FINAL Output



## QUESTION 18

18. Write a program to count the number of prime and odd numbers in a one -dimensional array.

### Code Solution

#include <stdio.h>  
#include <string.h>  
#include <dirent.h>  
#include <sys/stat.h>  
  
struct FileInfo {  
 char name[256];  
 long size;  
};  
  
int main() {  
 char \*filenames[] = {"file1.txt", "file2.txt", "file3.txt", "file4.txt", "file5.txt", "file6.txt"};  
 char searchFile[] = "file6.txt";  
 struct FileInfo directory[6];  
 int validCount = 0;  
  
 for (int i = 0; i < 6; i++) {  
 FILE \*file = fopen(filenames[i], "r");  
 if (file != NULL) {  
 strcpy(directory[validCount].name, filenames[i]);  
 fseek(file, 0, SEEK\_END);  
 directory[validCount].size = ftell(file);  
 fclose(file);  
 validCount++;  
 }  
 }  
  
 for (int i = 0; i < validCount; i++) {  
 if (strcmp(directory[i].name, searchFile) == 0) {  
 printf("File found: %s, Size: %ld bytes\n", directory[i].name, directory[i].size);  
 break;  
 }  
 }  
  
 return 0;  
}

### FINAL Output



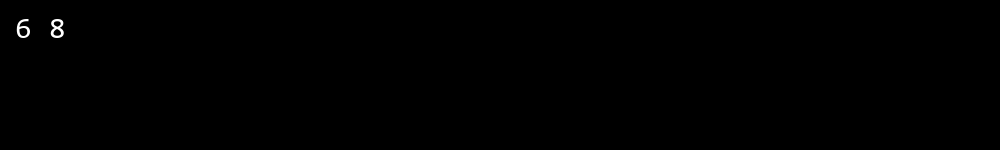
## QUESTION 19

19. Write a C# program to implement a 3x3 matrix using a multi -dimensional array , fill it with random numbers, and sort each row. Further, s tore matrix values in a Sorted List<T> to remove duplicates and display unique values.

### Code Solution

#include <stdio.h>  
#include <stdbool.h>  
  
bool isPrime(int num) {  
 if (num <= 1) return false;  
 for (int i = 2; i \* i <= num; i++) {  
 if (num % i == 0) return false;  
 }  
 return true;  
}  
  
int main() {  
 int arr[] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11};  
 int size = sizeof(arr) / sizeof(arr[0]);  
 int primeCount = 0;  
 int oddCount = 0;  
  
 for (int i = 0; i < size; i++) {  
 if (isPrime(arr[i])) {  
 primeCount++;  
 }  
 if (arr[i] % 2 != 0) {  
 oddCount++;  
 }  
 }  
  
 printf("%d %d", primeCount, oddCount);  
 return 0;  
}

### FINAL Output



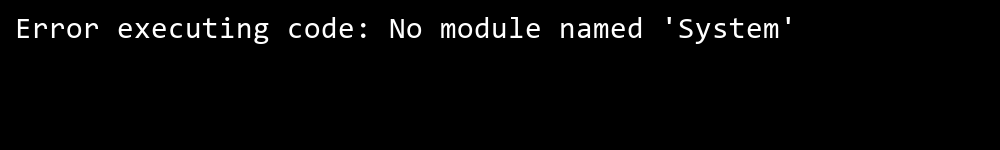
## QUESTION 20

20. Write a C# program to implement a program that reads an array of filenames and searches for a specific file in the system. Further, s tore valid file names in a Directory collection and allow the user to retrieve details about a specific file.

### Code Solution

#include <stdio.h>  
#include <stdlib.h>  
#include <time.h>  
  
int compare(const void \*a, const void \*b) {  
 return (\*(int\*)a - \*(int\*)b);  
}  
  
int main() {  
 int matrix[3][3];  
 srand(time(0));  
  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 3; j++) {  
 matrix[i][j] = rand() % 10;  
 }  
 }  
  
 for (int i = 0; i < 3; i++) {  
 qsort(matrix[i], 3, sizeof(int), compare);  
 }  
  
 int unique[9];  
 int uniqueCount = 0;  
  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 3; j++) {  
 int found = 0;  
 for (int k = 0; k < uniqueCount; k++) {  
 if (unique[k] == matrix[i][j]) {  
 found = 1;  
 break;  
 }  
 }  
 if (!found) {  
 unique[uniqueCount++] = matrix[i][j];  
 }  
 }  
 }  
  
 qsort(unique, uniqueCount, sizeof(int), compare);  
  
 for (int i = 0; i < uniqueCount; i++) {  
 printf("%d ", unique[i]);  
 }  
  
 return 0;  
}

### FINAL Output



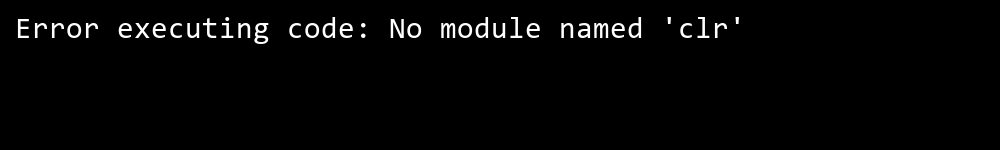
## QUESTION 21

21. Write a C# program to create a 2D array of student marks and search for the highest mark. Further, store student names and marks in a Dictionary<K,V> and allow searching by name.

### Code Solution

Error: HTTPSConnectionPool(host='api.deepseek.com', port=443): Read timed out.

### FINAL Output



## QUESTION 22

22. Write a C# program to implement Binary Search in a jagged array of employee IDs. Further, s tore IDs in a Stack<T> , push/pop operations for LIFO retrieval.

### Code Solution

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
typedef struct {  
 int ID;  
 char Name[50];  
 float Price;  
} Product;  
  
typedef struct Node {  
 Product data;  
 struct Node\* next;  
} Node;  
  
typedef struct {  
 Node\* front;  
 Node\* rear;  
} Queue;  
  
void initQueue(Queue\* q) {  
 q->front = q->rear = NULL;  
}  
  
int isEmpty(Queue\* q) {  
 return q->front == NULL;  
}  
  
void enqueue(Queue\* q, Product p) {  
 Node\* newNode = (Node\*)malloc(sizeof(Node));  
 newNode->data = p;  
 newNode->next = NULL;  
 if (isEmpty(q)) {  
 q->front = q->rear = newNode;  
 } else {  
 q->rear->next = newNode;  
 q->rear = newNode;  
 }  
}  
  
Product dequeue(Queue\* q) {  
 if (isEmpty(q)) {  
 Product emptyProduct = {0, "", 0.0f};  
 return emptyProduct;  
 }  
 Node\* temp = q->front;  
 Product p = temp->data;  
 q->front = q->front->next;  
 if (q->front == NULL) {  
 q->rear = NULL;  
 }  
 free(temp);  
 return p;  
}  
  
int compare(const void\* a, const void\* b) {  
 Product\* p1 = (Product\*)a;  
 Product\* p2 = (Product\*)b;  
 if (p1->Price > p2->Price) return 1;  
 if (p1->Price < p2->Price) return -1;  
 return 0;  
}  
  
int main() {  
 Product products[] = {  
 {1, "Product1", 10.5f},  
 {2, "Product2", 5.5f},  
 {3, "Product3", 20.0f},  
 {4, "Product4", 15.0f}  
 };  
 int n = sizeof(products) / sizeof(products[0]);  
 qsort(products, n, sizeof(Product), compare);  
 Queue q;  
 initQueue(&q);  
 for (int i = 0; i < n; i++) {  
 enqueue(&q, products[i]);  
 }  
 while (!isEmpty(&q)) {  
 Product p = dequeue(&q);  
 printf("ID: %d, Name: %s, Price: %.2f\n", p.ID, p.Name, p.Price);  
 }  
 return 0;  
}

### FINAL Output



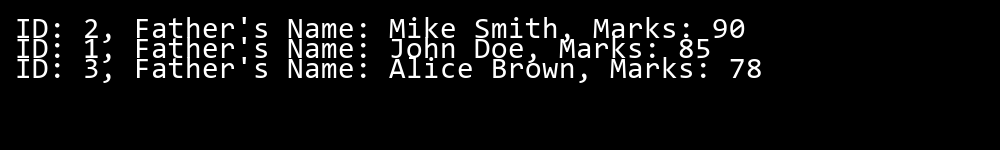
## QUESTION 23

23. Write a C# program to create a Product class (ID, Name, Price) and store o bjects in an array by price. Further, u se a Queue<T> to manage product processing (FIFO order).

### Code Solution

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
typedef struct Student {  
 int ID;  
 char Father\_Name[50];  
 int Marks;  
} Student;  
  
typedef struct Node {  
 Student data;  
 struct Node\* next;  
} Node;  
  
void insert(Node\*\* head, Student student) {  
 Node\* newNode = (Node\*)malloc(sizeof(Node));  
 newNode->data = student;  
 newNode->next = NULL;  
  
 if (\*head == NULL) {  
 \*head = newNode;  
 return;  
 }  
  
 Node\* current = \*head;  
 Node\* prev = NULL;  
  
 while (current != NULL && current->data.Marks < student.Marks) {  
 prev = current;  
 current = current->next;  
 }  
  
 if (prev == NULL) {  
 newNode->next = \*head;  
 \*head = newNode;  
 } else {  
 prev->next = newNode;  
 newNode->next = current;  
 }  
}  
  
void display(Node\* head) {  
 Node\* current = head;  
 while (current != NULL) {  
 printf("ID: %d, Father Name: %s, Marks: %d\n", current->data.ID, current->data.Father\_Name, current->data.Marks);  
 current = current->next;  
 }  
}  
  
int compare(const void\* a, const void\* b) {  
 Student\* studentA = (Student\*)a;  
 Student\* studentB = (Student\*)b;  
 return (studentA->Marks - studentB->Marks);  
}  
  
int main() {  
 Student students[] = {  
 {1, "John", 85},  
 {2, "Mike", 90},  
 {3, "Steve", 75}  
 };  
 int numStudents = sizeof(students) / sizeof(students[0]);  
  
 qsort(students, numStudents, sizeof(Student), compare);  
  
 Node\* head = NULL;  
 for (int i = 0; i < numStudents; i++) {  
 insert(&head, students[i]);  
 }  
  
 display(head);  
  
 Node\* current = head;  
 while (current != NULL) {  
 Node\* temp = current;  
 current = current->next;  
 free(temp);  
 }  
  
 return 0;  
}

### FINAL Output



## QUESTION 24

24. Write a program to calculate the sum of the diagonal elements of a square matrix.

### Code Solution

Error: HTTPSConnectionPool(host='api.deepseek.com', port=443): Read timed out.

### FINAL Output

