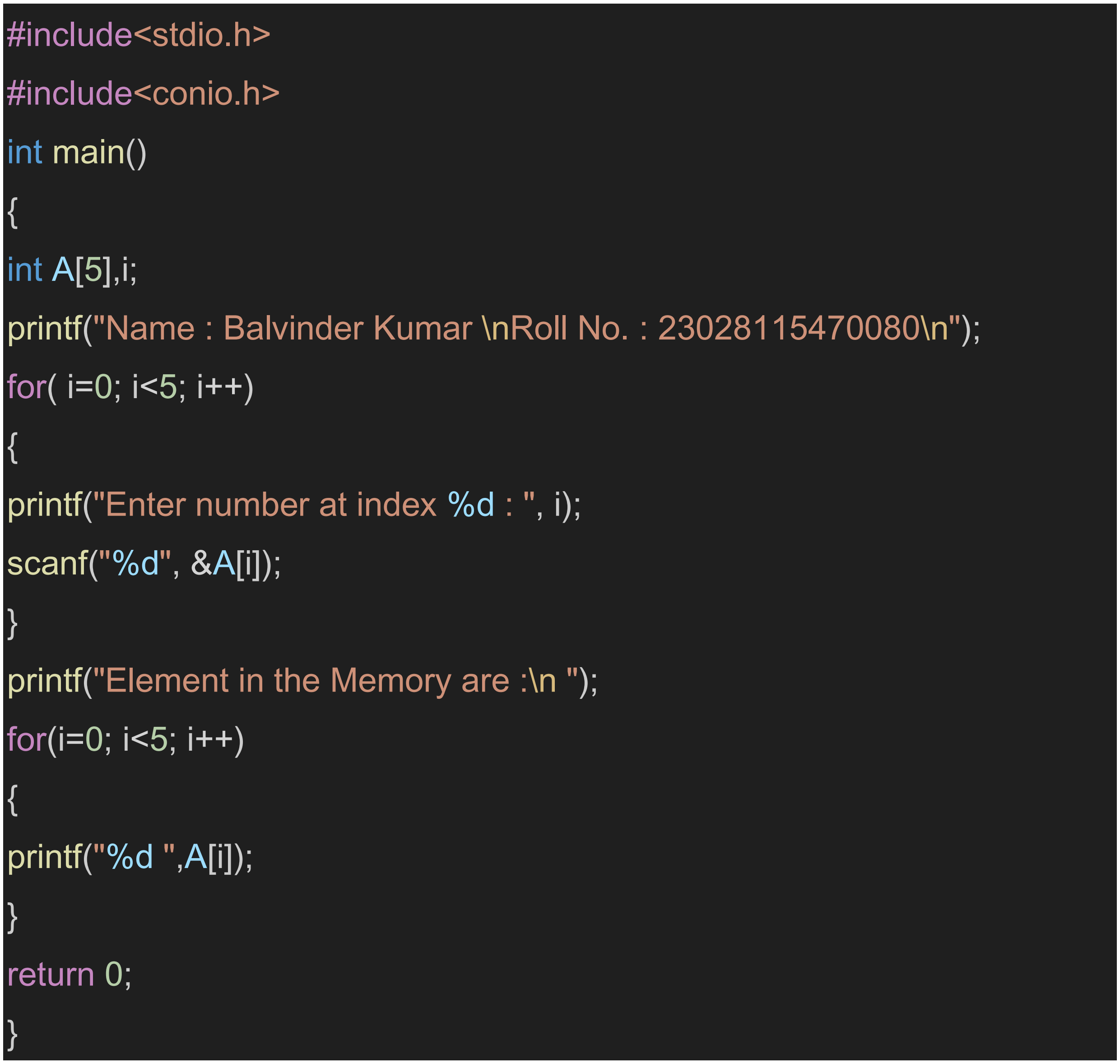
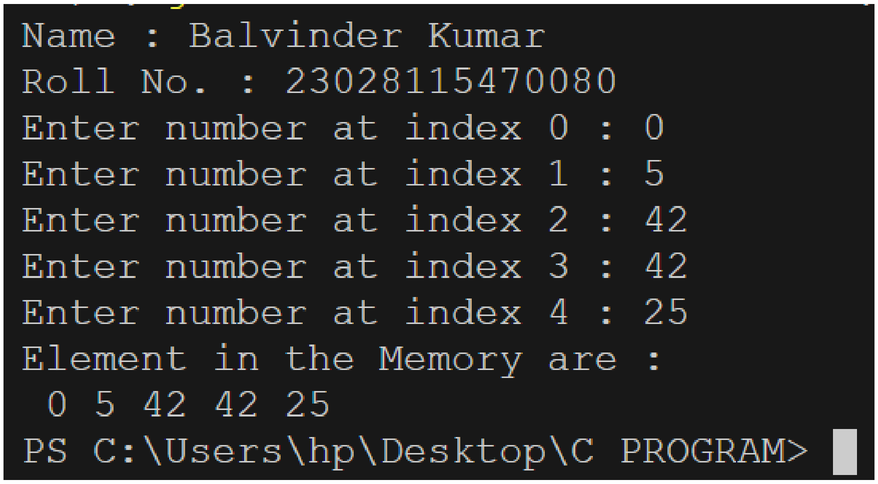
Program 1 : Write a program for inserting and displaying the content of Array.

Source Code :



```c
#include<stdio.h>
#include<conio.h>
int main()
{
int A[5],i;
printf("Name : Balvinder Kumar \nRoll No. : 23028115470080\n");
for( i=0; i<5; i++)
{
printf("Enter number at index %d : ", i);
scanf("%d", &A[i]);
}
printf("Element in the Memory are :\n ");
for(i=0; i<5; i++)
{
printf("%d ",A[i]);
}
return 0;
}
```

Output:



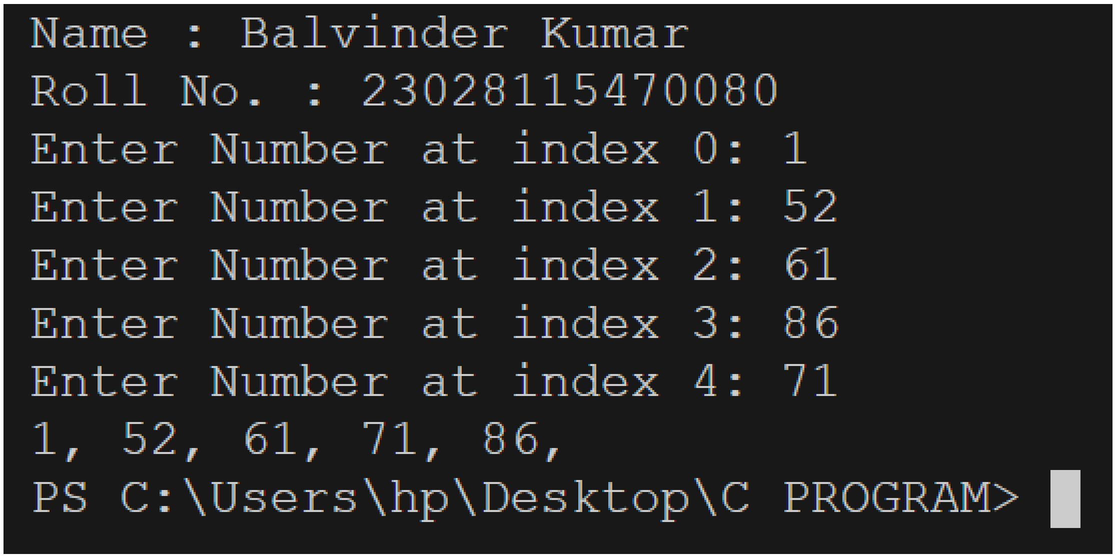
```
Name : Balvinder Kumar
Roll No. : 23028115470080
Enter number at index 0 : 0
Enter number at index 1 : 5
Enter number at index 2 : 42
Enter number at index 3 : 42
Enter number at index 4 : 25
Element in the Memory are :
 0 5 42 42 25
PS C:\Users\hp\Desktop\C PROGRAM> 
```

Program 2 : Write a program for sorting the elements of array using Bubble Sort.

Source Code :

```c
#include<stdio.h>
#include<conio.h>
int main()
{
    int A[5],i,j,t;
    printf("Name : Balvinder Kumar \nRoll No. :          23028115470080\n");
    for(i=0; i<5; i++)
    {
        printf("Enter Number at index %d: ",i);
        scanf("%d",&A[i]);
    }
    for(j=0; j<5; j++)
    {
        for(i=0; i<5-j; i++)
        {
            if(A[i] > A[i+1])
            {
                t = A[i];
                A[i] = A[i+1];
                A[i+1] = t;
            }
        }
    }
    for(i=0; i<5; i++ )
    {
        printf("%d, ", A[i]);
    }
    return 0;
}
```

Output:



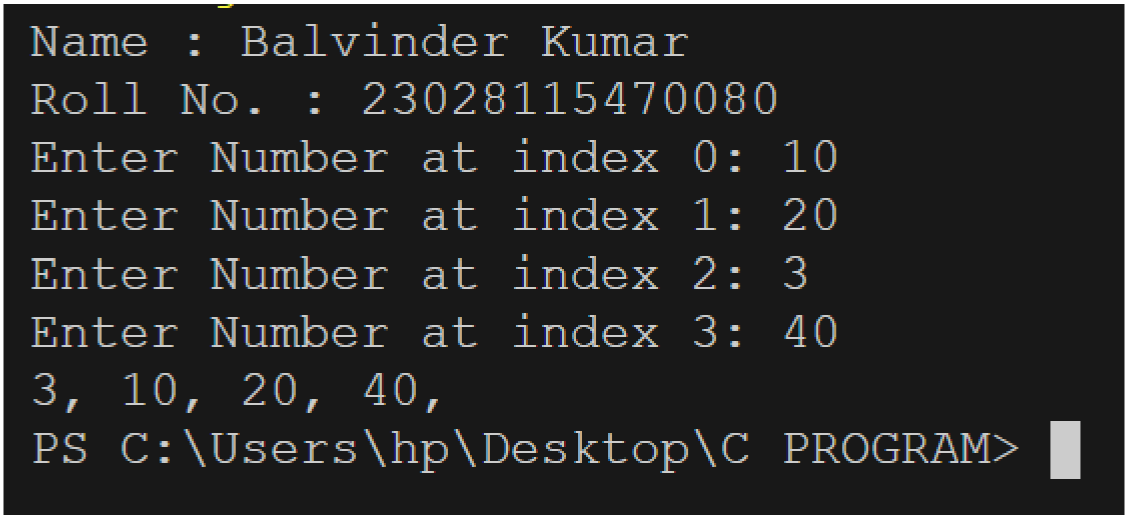
```
Name : Balvinder Kumar
Roll No. : 23028115470080
Enter Number at index 0: 1
Enter Number at index 1: 52
Enter Number at index 2: 61
Enter Number at index 3: 86
Enter Number at index 4: 71
1, 52, 61, 71, 86,
PS C:\Users\hp\Desktop\C PROGRAM>
```

Program 3 : Write a program for sorting the elements of array using Selection Sort.

Source Code :

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int A[4],i,j,small,loc;
    printf("Name : Balvinder Kumar \nRoll No. : 23028115470080\n");
    for(i=0; i<4; i++)
    {
        printf("Enter Number at index %d: ",i);
        scanf("%d",&A[i]);
    }
    for(j=0; j<4; j++)
    {
        small = A[j];
        loc = j;
        for(i=j+1; i<4; i++)
        {
            if(A[i] < small)
            {
                small = A[i];
                loc = i;
            }
        }
        if(loc != j)
        {
            A[loc] = A[j];
            A[j] = small;
        }
    }
    for(i=0; i<4; i++ )
    {
            printf("%d, ", A[i]);
    }
    return 0;
}
```

Output:



```
Name : Balvinder Kumar
Roll No. : 23028115470080
Enter Number at index 0: 10
Enter Number at index 1: 20
Enter Number at index 2: 3
Enter Number at index 3: 40
3, 10, 20, 40,
PS C:\Users\hp\Desktop\C PROGRAM>
```

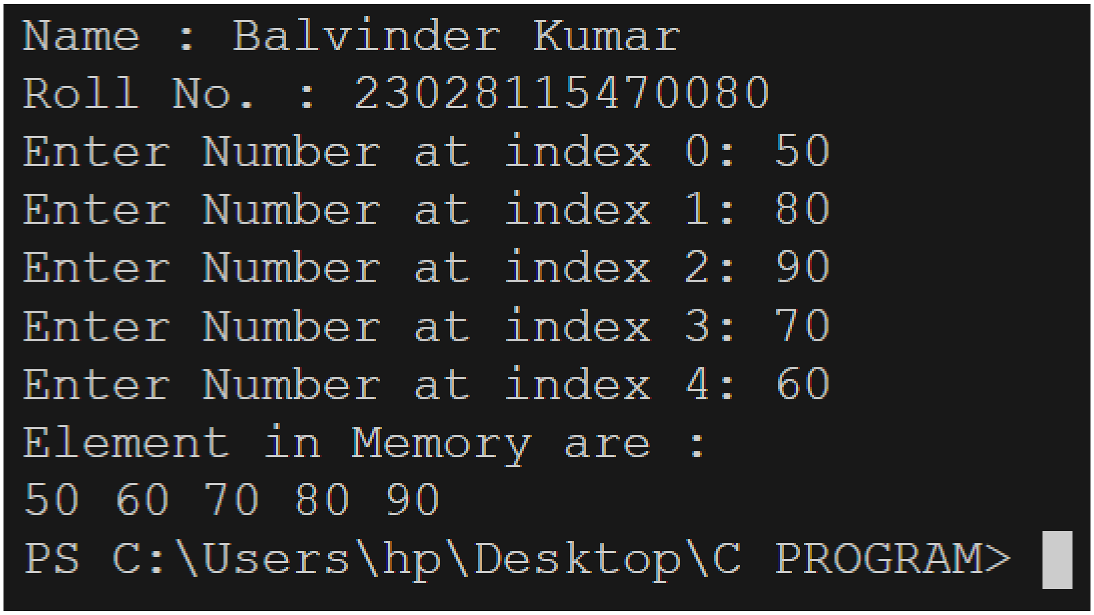
Program 4: Write a program for sorting the element of array using insertion sort.

Source Code:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int A[5],i,j,key;
    printf("Name : Balvinder Kumar \nRoll No. : 23028115470080\n");
    for(i=0; i<5; i++)
    {
        printf("Enter Number at index %d: ",i);
        scanf("%d",&A[i]);
    }
    for(i=1; i<5; i++)
    {
        key = A[i];
        j = i-1;
        while(j>=0 && A[j]>key)
        {
            A[j+1] = A[j];
            j = j-1;
        }
        A[j+1] = key;
    }
    printf("Element in Memory are :\n");
    for(i=0; i<5; i++)
    {
        printf("%d ", A[i]);
    }
```

```
  return 0;
}
```

Output:



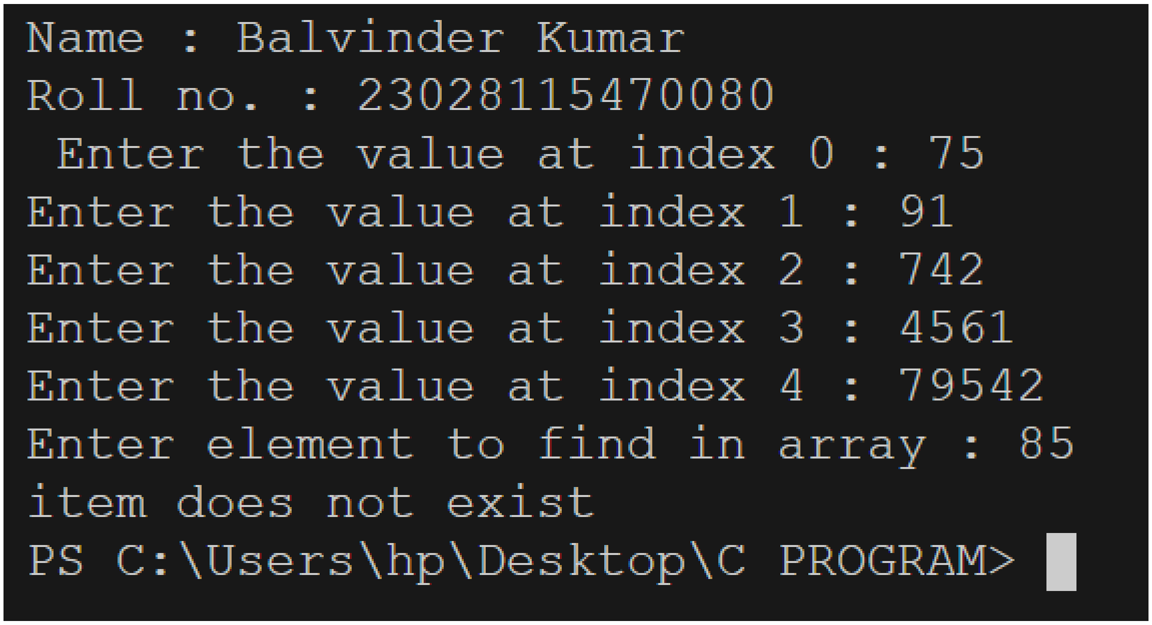
```
Name : Balvinder Kumar
Roll No. : 23028115470080
Enter Number at index 0: 50
Enter Number at index 1: 80
Enter Number at index 2: 90
Enter Number at index 3: 70
Enter Number at index 4: 60
Element in Memory are :
50 60 70 80 90
PS C:\Users\hp\Desktop\C PROGRAM>
```

Program 5 : Write a program for linear search in one dimensional array.

Source Code :

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int A[5], item, i;
    printf("Name : Balvinder Kumar \nRoll no. : 23028115470080\n ");
    for(i=0; i<5; i++)
    {
        printf("Enter the value at index %d : ",i);
        scanf("%d", &A[i]);
    }
    printf("Enter element to find in array : ");
    scanf("%d", &item);
    for(i=0; i<6 && A[i]!= item; i++);
    if(i>4)
    {
        printf("item does not exist");
    }
    else
    {
        printf("item found at index %d",i);
    }
    return 0;
}
```

Output:



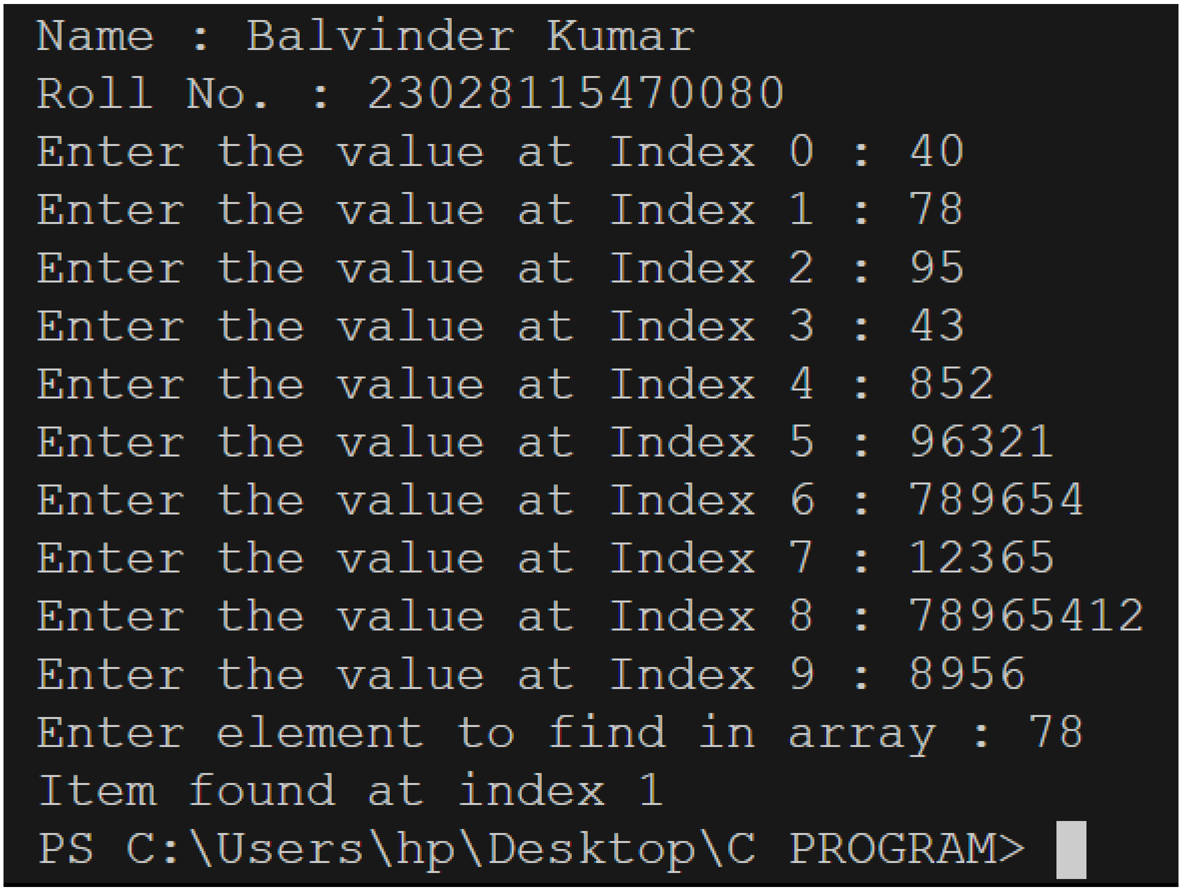
```
Name : Balvinder Kumar
Roll no. : 23028115470080
 Enter the value at index 0 : 75
Enter the value at index 1 : 91
Enter the value at index 2 : 742
Enter the value at index 3 : 4561
Enter the value at index 4 : 79542
Enter element to find in array : 85
item does not exist
PS C:\Users\hp\Desktop\C PROGRAM>
```

Program 6 : Write a program for finding an element in sorted array using Binary search.

Source Code :

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int A[10], beg=0, end=9, mid, item, i;
    mid =(beg +end)/2;
    printf("Name : Balvinder Kumar\nRoll No. : 23028115470080\n");
    for(i=0; i<10; i++)
    {
        printf("Enter the value at Index %d : ",i);
        scanf("%d", &A[i]);
    }
    printf("Enter element to find in array : ");
    scanf("%d", &item);
    while(beg<=end && A[mid]!= item)
    {
        if(item<A[mid])
            end = mid-1;
        else
            beg = mid + 1;
        mid = (beg + end)/2;
    }
    if(beg>end)
    {
        printf("item does not exist");
    }
    else
    {
        printf("Item found at index %d", mid);
    }
    return 0;
}
```

Output:



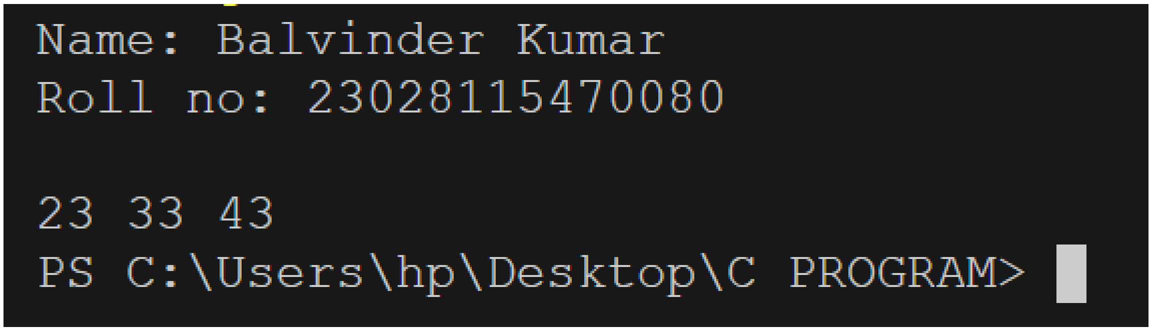
```
Name : Balvinder Kumar
Roll No. : 23028115470080
Enter the value at Index 0 : 40
Enter the value at Index 1 : 78
Enter the value at Index 2 : 95
Enter the value at Index 3 : 43
Enter the value at Index 4 : 852
Enter the value at Index 5 : 96321
Enter the value at Index 6 : 789654
Enter the value at Index 7 : 12365
Enter the value at Index 8 : 78965412
Enter the value at Index 9 : 8956
Enter element to find in array : 78
Item found at index 1
PS C:\Users\hp\Desktop\C PROGRAM>
```

Program 7: Write a program to implement the static one way link list.

Source code:-

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

typedef struct node {
    int info;
    struct node *link;
} Node;

int main() {
    Node *start, *n1, *n2, *n3, *n4, *ptr;

    printf("Name: Balvinder Kumar\nRoll no: 23028115470080\n");

    // Allocate memory for nodes
    n1 = (Node*)malloc(sizeof(Node));
    n2 = (Node*)malloc(sizeof(Node));
    n3 = (Node*)malloc(sizeof(Node));

    start = n1;

    n1->info = 23;
    n1->link = n2;

    n2->info = 33;
    n2->link = n3;

    n3->info = 43;
    n3->link = NULL;
    printf("\n");

    ptr = start;
    while (ptr != NULL) {
        printf("%d ", ptr->info);
        ptr = ptr->link;
    }

    // Free dynamically allocated memory
    free(n1);
    free(n2);
    free(n3);

    return 0;
}
```

Output:

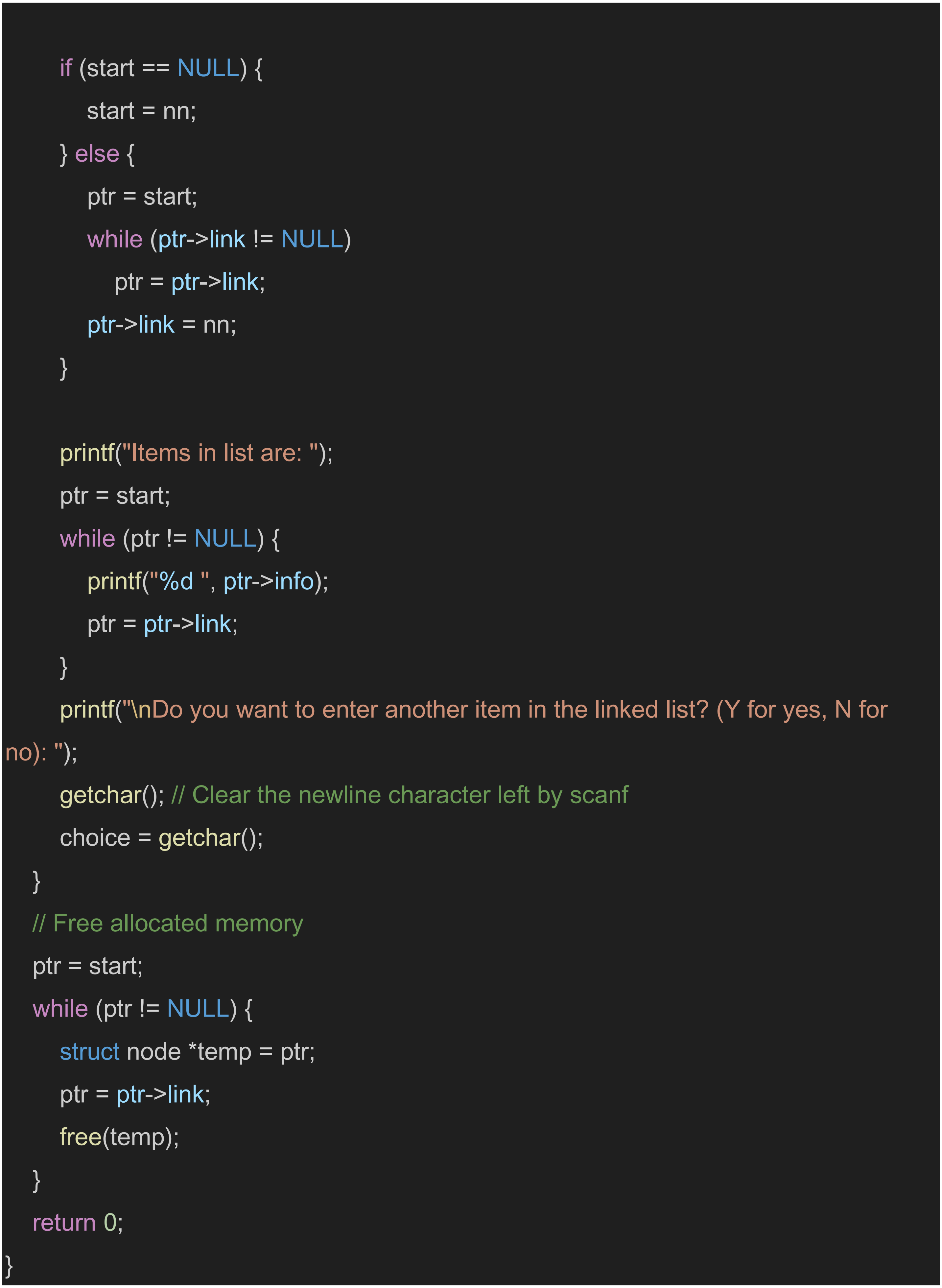


```
Name: Balvinder Kumar
Roll no: 23028115470080

23 33 43
PS C:\Users\hp\Desktop\C PROGRAM>
```

Program 8:- write a program insertion in the one way link list at the end.

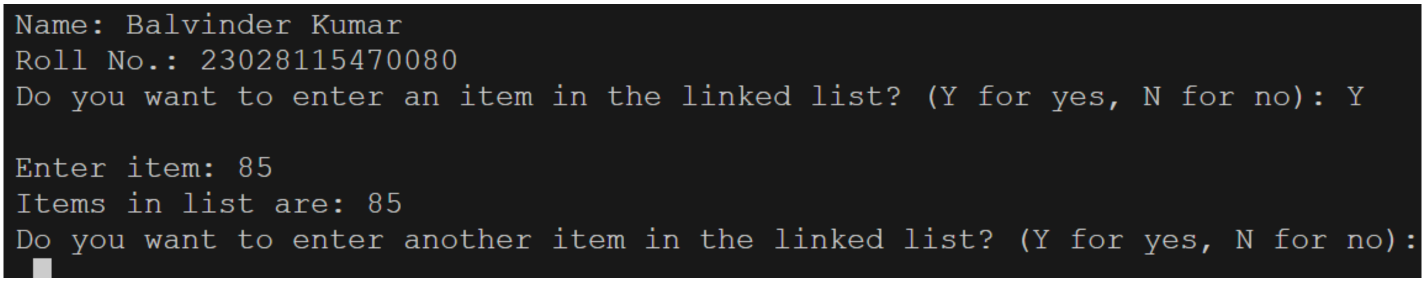
Source code:-

```
#include <stdio.h>

#include <stdlib.h>

struct node {
    int info;
    struct node *link;
};

int main() {
    struct node *start = NULL, *ptr = NULL, *nn = NULL;
    int item;
    char choice;

    printf("Name: Balvinder Kumar\nRoll No.: 230281154700080\n");
    printf("Do you want to enter an item in the linked list? (Y for yes, N for no): ");
    choice = getchar();

    while (choice == 'Y' || choice == 'y') {
        nn = malloc(sizeof(struct node));
        if (nn == NULL) {
            printf("\nMemory is full");
            return 1; // Exit with error
        }

        printf("\nEnter item: ");
        scanf("%d", &item);
        nn->info = item;
        nn->link = NULL;
```



```
        if (start == NULL) {
            start = nn;
        } else {
            ptr = start;
            while (ptr->link != NULL)
                ptr = ptr->link;
            ptr->link = nn;
        }

        printf("Items in list are: ");
        ptr = start;
        while (ptr != NULL) {
            printf("%d ", ptr->info);
            ptr = ptr->link;
        }
        printf("\nDo you want to enter another item in the linked list? (Y for yes, N for no): ");
        getchar(); // Clear the newline character left by scanf
        choice = getchar();
    }
    // Free allocated memory
    ptr = start;
    while (ptr != NULL) {
        struct node *temp = ptr;
        ptr = ptr->link;
        free(temp);
    }
    return 0;
}
```

Output:

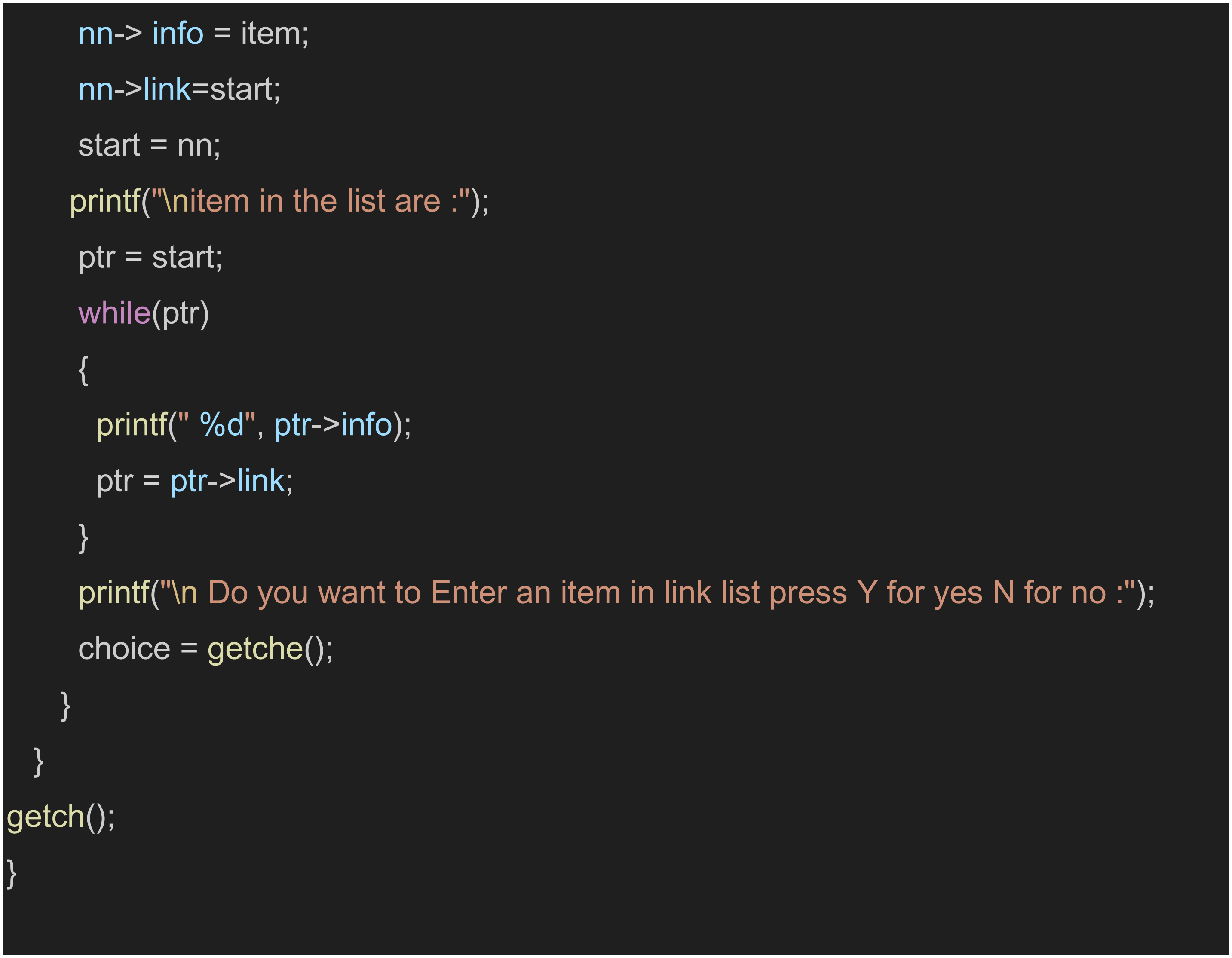


```
Name: Balvinder Kumar
Roll No.: 23028115470080
Do you want to enter an item in the linked list? (Y for yes, N for no): Y

Enter item: 85
Items in list are: 85
Do you want to enter another item in the linked list? (Y for yes, N for no):
```

Program 9: Write a program insertion in the one way link list at the beginning.

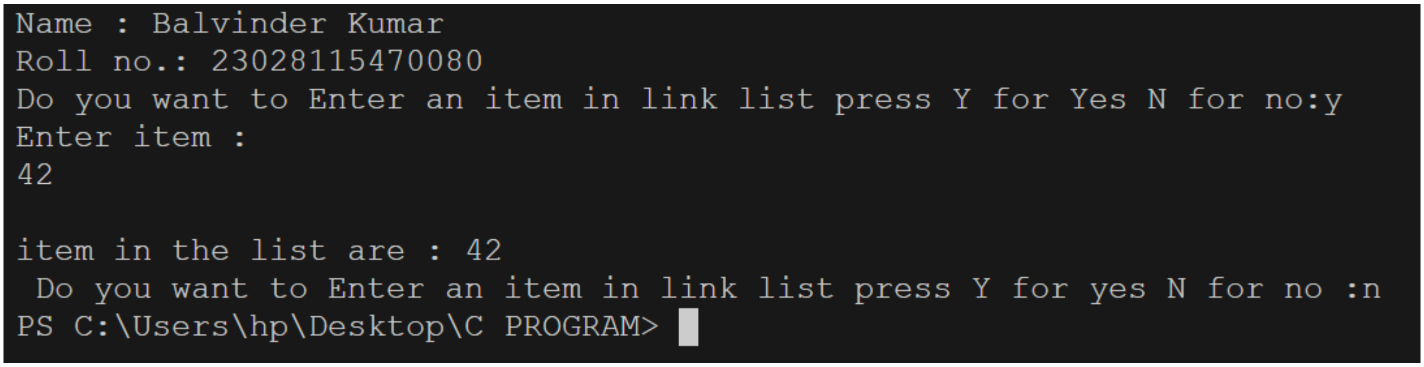
Source code :

```
#include<stdio.h>
#include<conio.h>
struct node
{
   int info;
   struct node*link;
};
void main ()
{
  struct node *start= NULL, *ptr = NULL, *nn = NULL;
  int item;
  char choice;
  printf("Name : Balvinder Kumar\n");
  printf("Roll no.: 23028115470080\n");
  printf("Do you want to Enter an item in link list press Y for Yes N for no:");
  choice = getche();
  while(choice == 'Y' || choice == 'y')
  {
    nn= malloc(sizeof(struct node));
    if (nn == NULL)
    {
     printf("\n Memory is full");
    }
    else
    {
     nn->link = NULL;
     printf("\nEnter item :");
     scanf("%d", &item);
```



```
        nn-> info = item;
        nn->link=start;
        start = nn;
       printf("\nitem in the list are :");
        ptr = start;
        while(ptr)
        {
          printf(" %d", ptr->info);
          ptr = ptr->link;
        }
        printf("\n Do you want to Enter an item in link list press Y for yes N for no :");
        choice = getche();
      }
   }
getch();
}
```

Output:



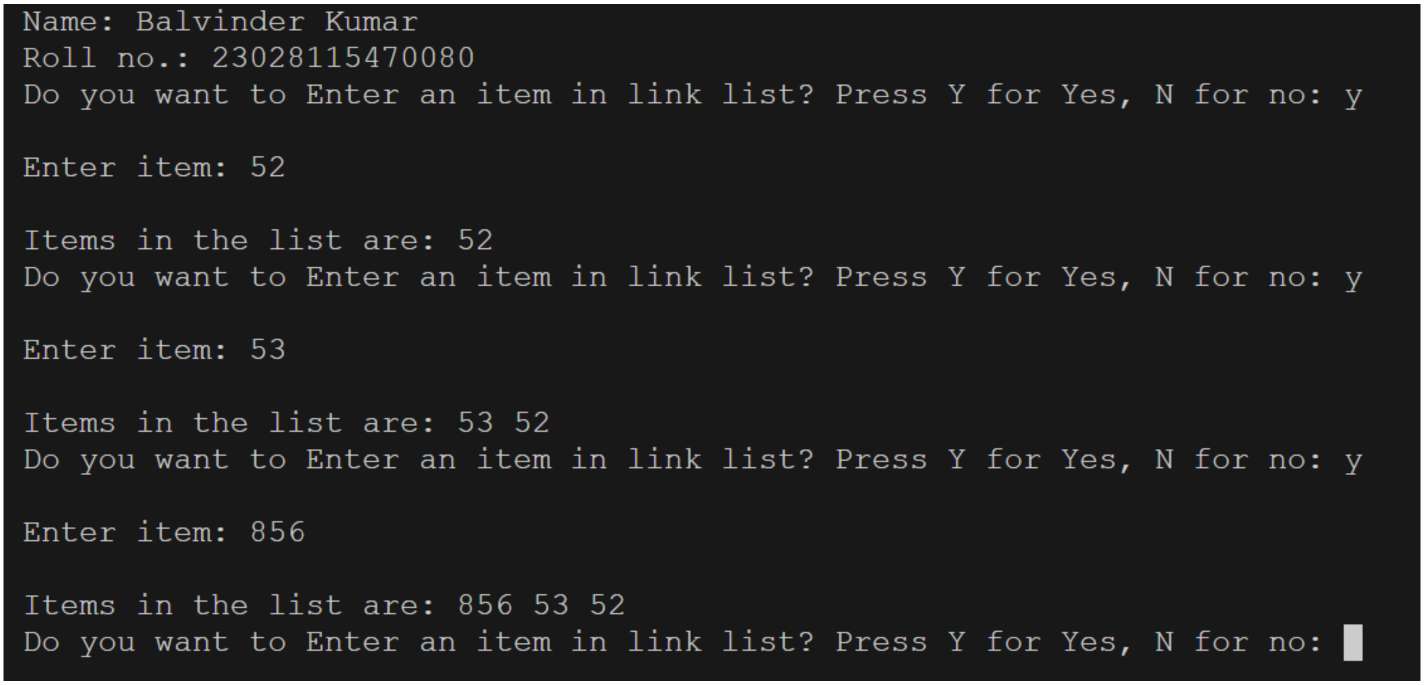
```
Name : Balvinder Kumar
Roll no.: 23028115470080
Do you want to Enter an item in link list press Y for Yes N for no:y
Enter item :
42

item in the list are : 42
 Do you want to Enter an item in link list press Y for yes N for no :n
PS C:\Users\hp\Desktop\C PROGRAM>
```

Program 10: Write a program insertion in one way ordered link list.

Source Code:

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

struct node {
    int info;
    struct node *link;
};

int main() {
    struct node *start = NULL, *ptr = NULL, *nn = NULL;
    int item;
    char choice;
    printf("Name : Balvinder Kumar\nRoll No. : 23028115470080\n");
    printf("Do you want to Enter an item in link list press Y(yes) or N(no)");
    choice = getchar();
    while (choice == 'Y' || choice == 'y') {
        nn = (struct node *)malloc(sizeof(struct node));
        if (nn == NULL) {
            printf("\nMemory is full");
            exit(1); // Exiting if memory allocation fails
        }
        else {
            nn->link = NULL;
            printf("\nEnter item: ");
            scanf("%d", &item);
            nn->info = item;
            nn->link = start;
            start = nn;
            printf("\nItem in the list are: ");
            ptr = start;
            while (ptr) {
                printf("%d ", ptr->info);
                ptr = ptr->link;
            }
            printf("\nDo you want to Enter an item in link list Y(yes) or N(no)");
            getchar(); // Consume the newline character left in the buffer by scanf
            choice = getchar();
        }
    }
    return 0;
}
```

Output:



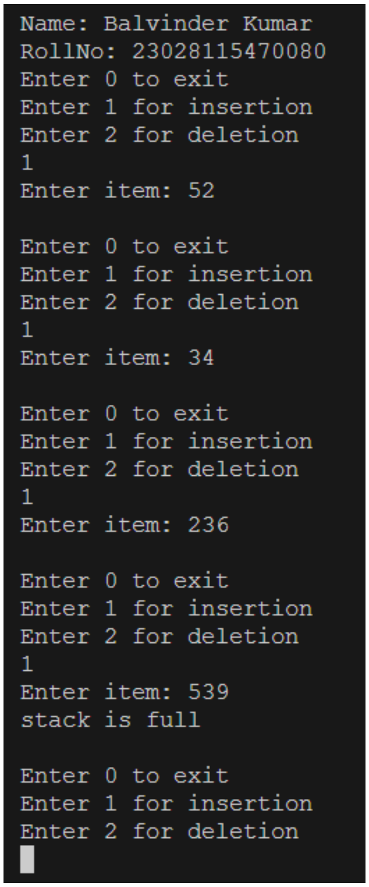
```
Name: Balvinder Kumar
Roll no.: 23028115470080
Do you want to Enter an item in link list? Press Y for Yes, N for no: y

Enter item: 52

Items in the list are: 52
Do you want to Enter an item in link list? Press Y for Yes, N for no: y

Enter item: 53

Items in the list are: 53 52
Do you want to Enter an item in link list? Press Y for Yes, N for no: y

Enter item: 856

Items in the list are: 856 53 52
Do you want to Enter an item in link list? Press Y for Yes, N for no:
```

Program 11: Write a program to implement stack.

Source Code:

```
#include <stdio.h>

#define size 3

int top = -1;
int stack[size];

int pop() {
    if (top < 0) {
        printf("\nstack is empty\n");
        return -1; // return -1 to indicate error
    }
    return stack[top--];
}

void push(int item) {
    if (top >= size - 1) {
        printf("stack is full\n");
        return;
    }
    stack[++top] = item;
}

int main() {
    int choice, item;

    printf("\nName: Balvinder Kumar \nRollNo: 23028115470080\n");
    printf("Enter 0 to exit\n");
    printf("Enter 1 for insertion\n");
    printf("Enter 2 for deletion\n");

    scanf("%d", &choice);

    while (choice != 0) {
        switch (choice) {
            case 1:
                printf("Enter item: ");
                scanf("%d", &item);
                push(item);
                break;
            case 2:
                item = pop();
                if (item != -1)
                    printf("Item %d deleted\n", item);
                break;
            default:
                printf("Invalid selection\n");
        }

        printf("\nEnter 0 to exit\n");
        printf("Enter 1 for insertion\n");
        printf("Enter 2 for deletion\n");
        scanf("%d", &choice);
    }

    return 0;
}
```

Output:



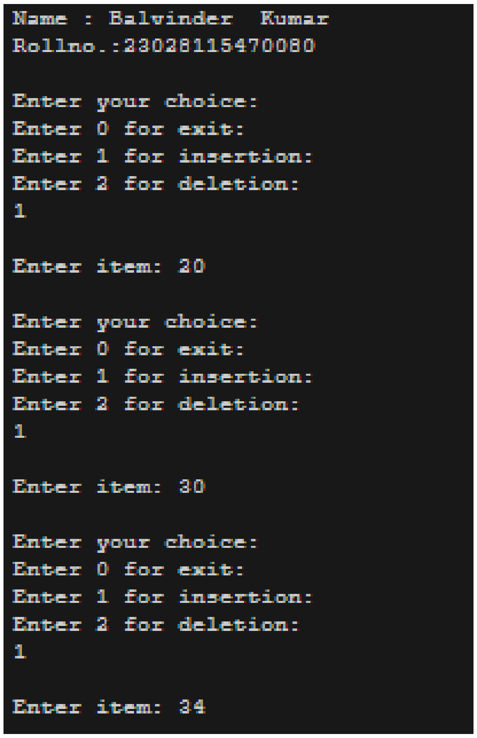
```
Name: Balvinder Kumar
RollNo: 23028115470080
Enter 0 to exit
Enter 1 for insertion
Enter 2 for deletion
1
Enter item: 52

Enter 0 to exit
Enter 1 for insertion
Enter 2 for deletion
1
Enter item: 34

Enter 0 to exit
Enter 1 for insertion
Enter 2 for deletion
1
Enter item: 236

Enter 0 to exit
Enter 1 for insertion
Enter 2 for deletion
1
Enter item: 539
stack is full

Enter 0 to exit
Enter 1 for insertion
Enter 2 for deletion
```

Program 12: Write a program to insertion of circular queue.

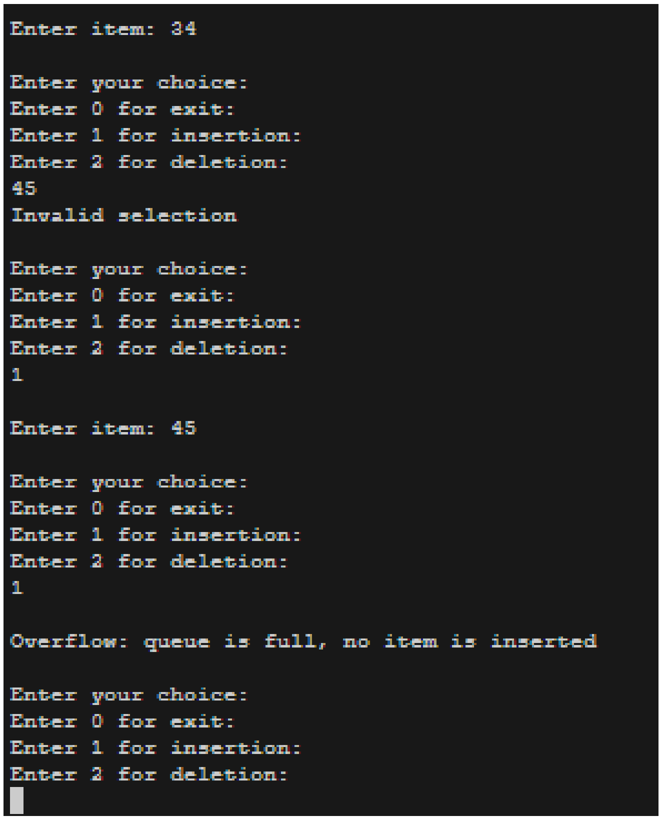
Source code :

```
#include <stdio.h>
#include <stdlib.h> // Include this for using exit()

#define size 4

int front = -1, rear = -1, item;
int queue[size];

int isfull() {
    if ((rear + 1) % size == front)
        return 1;
    else
        return 0;
}

int isempty() {
    if (front == -1 && rear == -1)
        return 1;
    else
        return 0;
}

void dequeue() {
    if (isempty()) {
        printf("\nUnderflow: queue is empty, no item is deleted\n");
    } else {
        item = queue[front];
        printf("\nitem %d is deleted from the queue\n", item);
        if (front == rear)
            front = rear = -1;
        else
            front = (front + 1) % size;
    }
}

void enqueue() {
    if (isfull()) {
        printf("\nOverflow: queue is full, no item is inserted\n");
    } else {
        printf("\nEnter item: ");
        scanf("%d", &item);
        if (front == -1 && rear == -1) {
            front = 0;
            rear = 0;
        } else {
            rear = (rear + 1) % size;
        }
        queue[rear] = item;
    }
}

int main() {
    int choice;

    printf("Name : Balvinder  Kumar\n");
    printf("Rollno.:23028115470080\n");

    while (1) {
        printf("\nEnter your choice:\n");
        printf("Enter 0 for exit:\n");
        printf("Enter 1 for insertion:\n");
        printf("Enter 2 for deletion:\n");

        scanf("%d", &choice);

        switch (choice) {
            case 0:
                return 0; // exit from the main function
            case 1:
                enqueue();
                break;
            case 2:
                dequeue();
                break;
            default:
                printf("Invalid selection\n");
        }
    }

    return 0;
}
```

Output:



```
Name : Balvinder  Kumar
Rollno.:23028115470080

Enter your choice:
Enter 0 for exit:
Enter 1 for insertion:
Enter 2 for deletion:
1

Enter item: 20

Enter your choice:
Enter 0 for exit:
Enter 1 for insertion:
Enter 2 for deletion:
1

Enter item: 30

Enter your choice:
Enter 0 for exit:
Enter 1 for insertion:
Enter 2 for deletion:
1

Enter item: 34
```



```
Enter item: 34

Enter your choice:
Enter 0 for exit:
Enter 1 for insertion:
Enter 2 for deletion:
45
Invalid selection

Enter your choice:
Enter 0 for exit:
Enter 1 for insertion:
Enter 2 for deletion:
1

Enter item: 45

Enter your choice:
Enter 0 for exit:
Enter 1 for insertion:
Enter 2 for deletion:
1

Overflow: queue is full, no item is inserted

Enter your choice:
Enter 0 for exit:
Enter 1 for insertion:
Enter 2 for deletion:
```

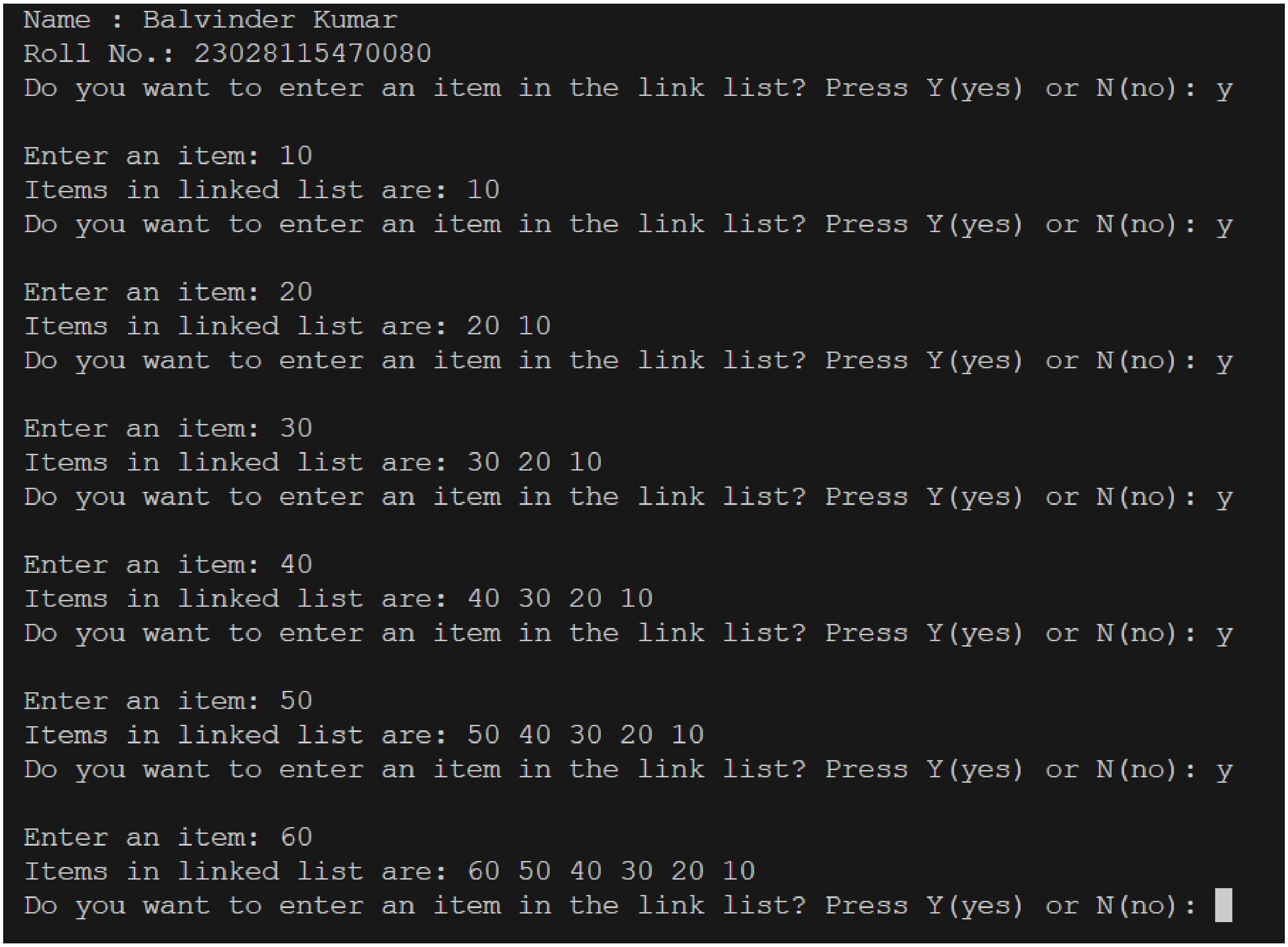
Program 13: write a program to insertion at the beginning using two way link list.

Source Code:

```c
#include<stdio.h>
#include<stdlib.h> // Include this for using malloc()

struct node {
    struct node *prev;
    int info;
    struct node *next;
};

int main() {
    struct node *head = NULL, *tail, *ptr, *nn = NULL;
    int item;
    char choice;

    printf("Name : Balvinder Kumar\nRoll No.: 23028115470080\n");
    printf("Do you want to enter an item in the link list? Press Y(yes) or N(no): ");

    scanf(" %c", &choice); // Use space before %c to consume any whitespace characters

    while (choice == 'Y' || choice == 'y') {
        nn = (struct node*)malloc(sizeof(struct node));

        if (nn == NULL) {
            printf("\nOverflow\n");
            exit(1); // Exit the program if memory allocation fails
        }
        else {
            nn->prev = NULL;
            nn->next = NULL;
            printf("\nEnter an item: ");
            scanf("%d", &item);
            nn->info = item;

            if (head == NULL) {
                head = nn;
                tail = nn;
            }
            else {
                nn->next = head;
                head->prev = nn;
                head = nn;
            }
        }

        ptr = head;
        printf("Items in linked list are: ");
        while (ptr != NULL) {
            printf("%d ", ptr->info); // Print a space after each item
            ptr = ptr->next;
        }

        printf("\nDo you want to enter an item in the link list? Press Y(yes) or N(no): ");
        scanf(" %c", &choice); // Use space before %c to consume any whitespace
characters
    }
```

```
   return 0;
}
```

Output:



```
Name : Balvinder Kumar
Roll No.: 23028115470080
Do you want to enter an item in the link list? Press Y(yes) or N(no): y

Enter an item: 10
Items in linked list are: 10
Do you want to enter an item in the link list? Press Y(yes) or N(no): y

Enter an item: 20
Items in linked list are: 20 10
Do you want to enter an item in the link list? Press Y(yes) or N(no): y

Enter an item: 30
Items in linked list are: 30 20 10
Do you want to enter an item in the link list? Press Y(yes) or N(no): y

Enter an item: 40
Items in linked list are: 40 30 20 10
Do you want to enter an item in the link list? Press Y(yes) or N(no): y

Enter an item: 50
Items in linked list are: 50 40 30 20 10
Do you want to enter an item in the link list? Press Y(yes) or N(no): y

Enter an item: 60
Items in linked list are: 60 50 40 30 20 10
Do you want to enter an item in the link list? Press Y(yes) or N(no):
```

Program 14: Write a program to implement the dynamic stack.

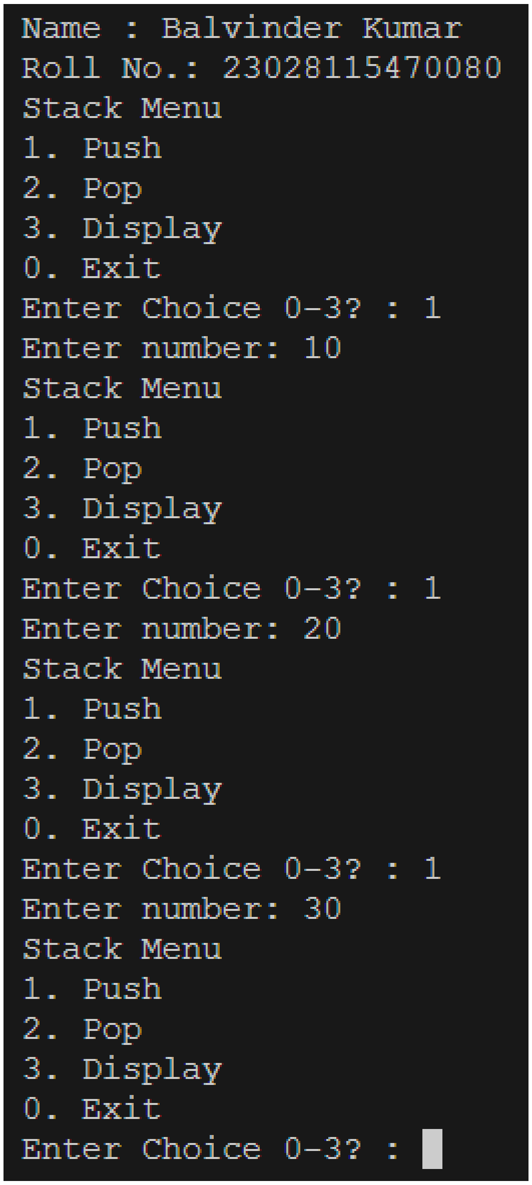
Source Code:

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

struct node
{
    int data;
    struct node *next;
};

struct node *top = NULL;

void display();
void push(int);
void pop();

int main()
{
    int n, ch;
    printf("Name : Balvinder Kumar\nRoll No.: 23028115470080\n");
    do
    {
        printf("Stack Menu\n1. Push\n2. Pop\n3. Display\n0. Exit\n");
        printf("Enter Choice 0-3? : ");
        scanf("%d", &ch);
        switch (ch)
        {
        case 1:
            printf("Enter number: ");
```

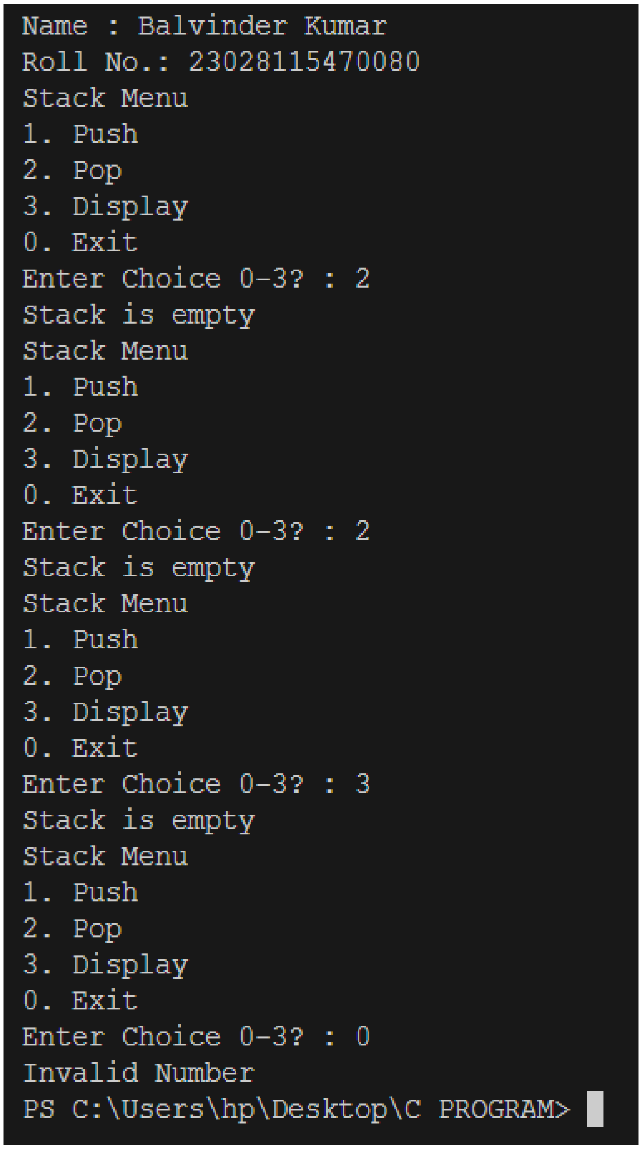
```c
            scanf("%d", &n);
            push(n);
            break;
        case 2:
            pop();
            break;
        case 3:
            display();
            break;
        default:
            printf("Invalid Number\n");
        }
    } while (ch != 0);

    return 0;
}

void push(int item)
{
    struct node *nptr = malloc(sizeof(struct node));
    if (nptr == NULL)
    {
        printf("Memory allocation failed\n");
        return;
    }
    nptr->data = item;
    nptr->next = top;
    top = nptr;
}

void display()
```

```
{
    struct node *temp = top;
    if (temp == NULL)
    {
        printf("Stack is empty\n");
        return;
    }
    printf("Stack elements: ");
    while (temp != NULL)
    {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}

void pop()
{
    if (top == NULL)
    {
        printf("Stack is empty\n");
        return;
    }
    struct node *temp = top;
    printf("%d deleted\n", temp->data);
    top = top->next;
    free(temp);
}
```

Output:



```
Name : Balvinder Kumar
Roll No.: 23028115470080
Stack Menu
1. Push
2. Pop
3. Display
0. Exit
Enter Choice 0-3? : 1
Enter number: 10
Stack Menu
1. Push
2. Pop
3. Display
0. Exit
Enter Choice 0-3? : 1
Enter number: 20
Stack Menu
1. Push
2. Pop
3. Display
0. Exit
Enter Choice 0-3? : 1
Enter number: 30
Stack Menu
1. Push
2. Pop
3. Display
0. Exit
Enter Choice 0-3? :
```

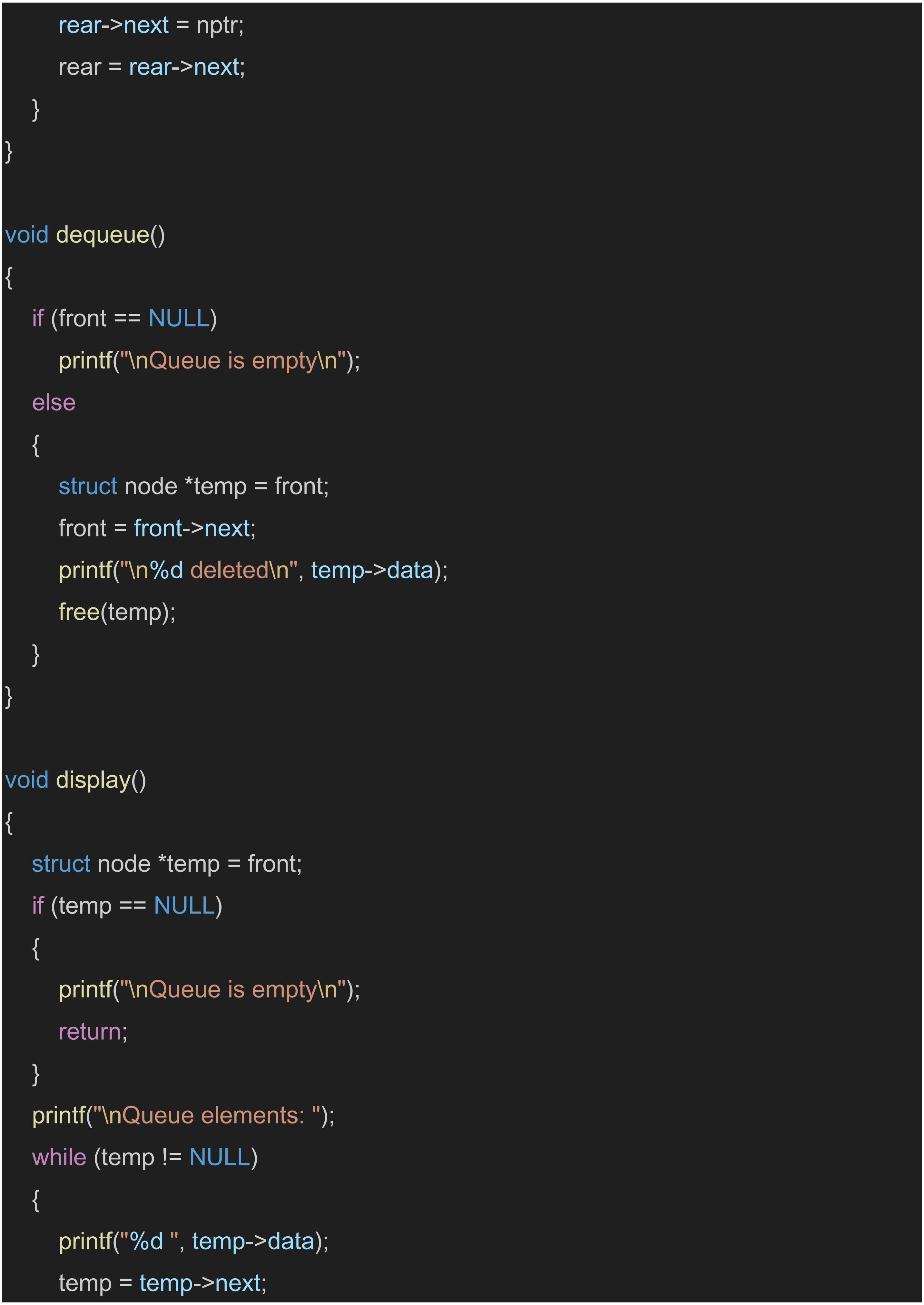


```
Name : Balvinder Kumar
Roll No.: 23028115470080
Stack Menu
1. Push
2. Pop
3. Display
0. Exit
Enter Choice 0-3? : 2
Stack is empty
Stack Menu
1. Push
2. Pop
3. Display
0. Exit
Enter Choice 0-3? : 2
Stack is empty
Stack Menu
1. Push
2. Pop
3. Display
0. Exit
Enter Choice 0-3? : 3
Stack is empty
Stack Menu
1. Push
2. Pop
3. Display
0. Exit
Enter Choice 0-3? : 0
Invalid Number
PS C:\Users\hp\Desktop\C PROGRAM>
```

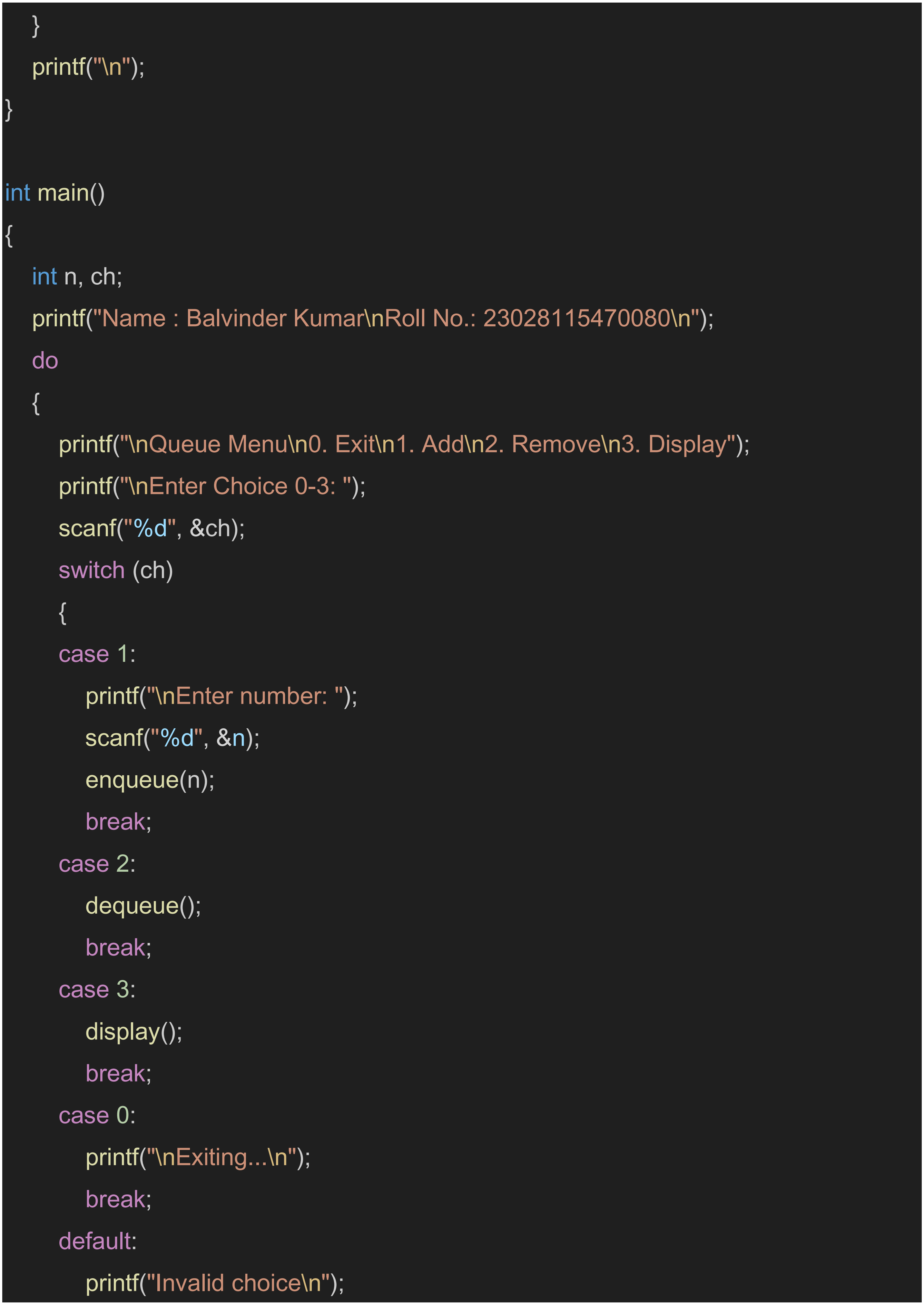
Program 16: Write a program to implementation of Dynamic Queue.

Soruce Code:

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

struct node
{
    int data;
    struct node *next;
};

struct node *front = NULL, *rear = NULL;

void enqueue(int item)
{
    struct node *nptr = malloc(sizeof(struct node));
    if (nptr == NULL)
    {
        printf("Memory allocation failed\n");
        return;
    }
    nptr->data = item;
    nptr->next = NULL;
    if (rear == NULL)
    {
        front = nptr;
        rear = nptr;
    }
    else
    {
```

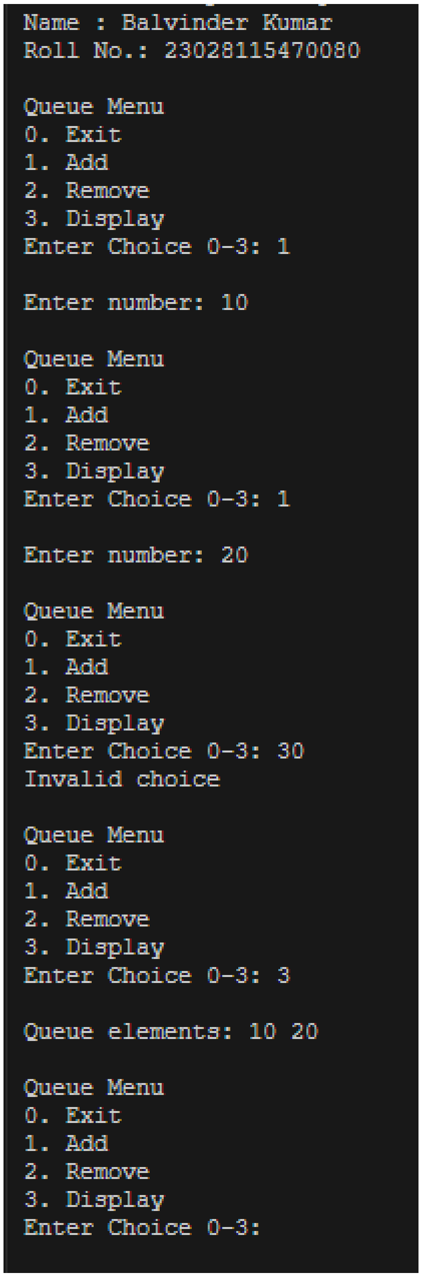


```
        rear->next = nptr;
        rear = rear->next;
    }
}

void dequeue()
{
    if (front == NULL)
        printf("\nQueue is empty\n");
    else
    {
        struct node *temp = front;
        front = front->next;
        printf("\n%d deleted\n", temp->data);
        free(temp);
    }
}

void display()
{
    struct node *temp = front;
    if (temp == NULL)
    {
        printf("\nQueue is empty\n");
        return;
    }
    printf("\nQueue elements: ");
    while (temp != NULL)
    {
        printf("%d ", temp->data);
        temp = temp->next;
```



```
    }
    printf("\n");
}

int main()
{
    int n, ch;
    printf("Name : Balvinder Kumar\nRoll No.: 23028115470080\n");
    do
    {
        printf("\nQueue Menu\n0. Exit\n1. Add\n2. Remove\n3. Display");
        printf("\nEnter Choice 0-3: ");
        scanf("%d", &ch);
        switch (ch)
        {
        case 1:
            printf("\nEnter number: ");
            scanf("%d", &n);
            enqueue(n);
            break;
        case 2:
            dequeue();
            break;
        case 3:
            display();
            break;
        case 0:
            printf("\nExiting...\n");
            break;
        default:
            printf("Invalid choice\n");
        }
    } while (ch != 0);


    return 0;
}
```

Output:



```
Name : Balvinder Kumar
Roll No.: 23028115470080

Queue Menu
0. Exit
1. Add
2. Remove
3. Display
Enter Choice 0-3: 1

Enter number: 10

Queue Menu
0. Exit
1. Add
2. Remove
3. Display
Enter Choice 0-3: 1

Enter number: 20

Queue Menu
0. Exit
1. Add
2. Remove
3. Display
Enter Choice 0-3: 30
Invalid choice

Queue Menu
0. Exit
1. Add
2. Remove
3. Display
Enter Choice 0-3: 3

Queue elements: 10 20

Queue Menu
0. Exit
1. Add
2. Remove
3. Display
Enter Choice 0-3:
```