NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

PROGRAM: SOFTWARE ENGINEERING

DATA STRUCTURES LAB

LAB TASK-03

SUBMITTED BY:

Name: Ahmed Ali

Roll No: 22P-9318

INSTRUCTOR NAME: Sir Saood Sarwar

A DEPARTMENT OF COMPUTER SCIENCE

Q1 CODE:

```
#include<iostream>
using namespace std;

class list
{
	private:
		int size;
		int *arr;
		int *curr;
		int *pos;

	public:
		list(int size)
		{
			this->size=size;
			arr=new int[size];
			curr=nullptr;
			pos=nullptr;
		}

		void start()
		{
			curr=arr;
		}

		void tail()
		{
        curr=arr+size-1;
}

void next()
{
        if(curr!=nullptr && curr<arr+size-1)
        {
                curr++;
        }
}

void back()
{
        if(curr!=nullptr && curr>arr)
        {
                curr--;
        }
}

bool is_full()
{
        return pos!=nullptr && pos==arr+size-1;
}

bool is_empty()
{
        return pos==nullptr;
}
```

```
    void clear()
    {
        curr=nullptr;
        pos=nullptr;
    }

    int length()
    {
if(pos==nullptr)
        {
            return 0;
        }
        return(pos-arr)+1;
    }

    void print()
    {
        if(is_empty())
        {
            cout<<"List is empty"<<endl;
        }
        else
        {
            cout<<"list: ";
            for(int *ptr=arr; ptr<=pos; ptr++)
            {
                cout<<*ptr<<" ";
            }
            cout<<endl;
        }
}

void insert(int book_id, int position)
{
        if(is_full())
        {
                cout<<"Sorry! cannot add more, array is full!"<<endl;
                return;
        }
        if(position<0 || position>length())
        {
                cout<<"invalid position!"<<endl;
                return;
        }

        if(pos==nullptr)
        {
                pos=arr;
                *pos=book_id;
                curr=pos;
                print();
                return;
        }

        for(int *ptr=arr+length(); ptr>arr+position; ptr--)
        {
                *ptr=*(ptr-1);
        }
        *(arr+position)=book_id;

        pos++;

        print();

}


void remove(int position)

{

        if(is_empty())

        {

                cout<<"array is empty!"<<endl;

                return;

        }

        if(position<0 || position>=length())

        {

                cout<<"invalid position!"<<endl;

                return;

        }


        for(int *ptr=arr+position; ptr<arr+length()-1; ptr++)

        {

                *ptr=*(ptr+1);

        }

        pos--;

        if(length()==0)

        {

                pos=nullptr;

                curr=nullptr;

        }

        print();
```

```
}

int get(int position)
{
	if(position<0 || position>=length())
	{
		cout<<"invalid position!"<<endl;
		return -1;
	}
	return *(arr+position);
}

void update(int book_id, int position)
{
	if(position<0 || position>=length())
	{
		cout<<"invalid position!"<<endl;
		return;
	}
	*(arr+position)=book_id;
	print();
}

bool find(int book_id)
{
	for(int *ptr=arr; ptr<=pos; ptr++)
	{
		if(*ptr==book_id)
		{
```

```
                        return true;
                }
        }
        return false;
}

void reverse()
{
        for(int *ptr=arr, *i=pos; ptr<i; ptr++, i--)
        {
                int temp=*ptr;
                *ptr=*i;
                *i=temp;
        }
        print();
}

void sort()
{
        for(int *ptr=arr; ptr<pos; ptr++)
        {
                for(int *i=ptr+1; i<=pos; i++)
                {
                        if(*ptr>*i)
                        {
                                int temp=*ptr;
                                *ptr=*i;
                                *i=temp;
                        }
```

```
                                        }
                                }
                                print();
                        }

                        ~list()
                        {
                                delete []arr;
                        }
};

int main()
{
        int size, choice, book_id, position;
        cout<<"Enter size of list: ";
        cin>>size;
        list List(size);
        do
        {
                cout<<endl<<"Menu: "<<endl;
                cout<<"1. Insert ID of book: "<<endl;
                cout<<"2. Remove ID of book: "<<endl;
                cout<<"3. Update ID of book: "<<endl;
                cout<<"4. Find ID of book: "<<endl;
                cout<<"5. Print the list"<<endl;
                cout<<"6. Reverse the list"<<endl;
                cout<<"7. Sort list"<<endl;
                cout<<"8. Clear the list"<<endl;
                cout<<"9. Check if full or not"<<endl;
```

```
cout<<"10. Check if empty or not"<<endl;

cout<<"11. Get book ID: "<<endl;

cout<<"12. Exit"<<endl;

cout<<"Enter your choice: ";

cin>>choice;


switch(choice)

{

case 1:

        cout<<"Enter book ID to insert: ";

        cin>>book_id;

        cout<<"Enter position to insert: ";

        cin>>position;

        List.insert(book_id, position);

        break;


case 2:

        cout<<"Enter position to remove: ";

        cin>>position;

        List.remove(position);

        break;


case 3:

        cout<<"Enter new book ID: ";

        cin>>book_id;

        cout<<"Enter position to update: ";

        cin>>position;

        List.update(book_id, position);

        break;
```

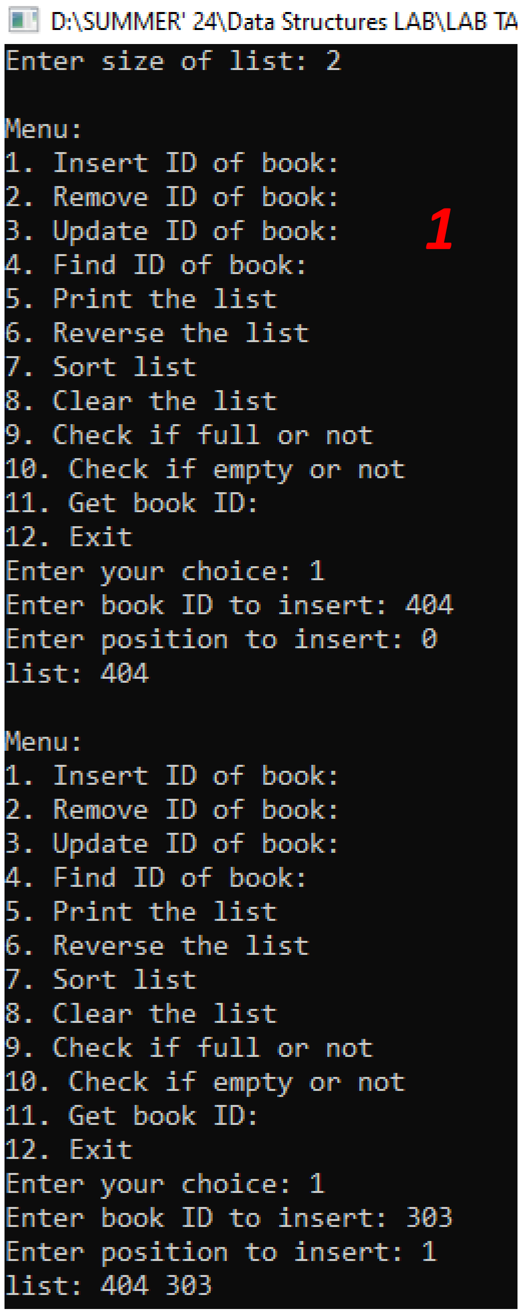
```
case 4:
        cout<<"Enter book ID to find: ";
        cin>>book_id;
        if(List.find(book_id))
        {
                cout<<"Book ID found in the list"<<endl;
        }
        else
        {
                cout<<"Book ID not found in the list"<<endl;
        }
        break;

case 5:
        List.print();
        break;

case 6:
        List.reverse();
        break;

case 7:
        List.sort();
        break;

case 8:
        List.clear();
        cout<<"List cleared"<<endl;
```

```
                break;

        case 9:
                if(List.is_full())
                {
                        cout<<"List is full"<<endl;
                }
                else
                {
                        cout<<"List is not full"<<endl;
                }
                break;

        case 10:
                if(List.is_empty())
                {
                        cout<<"List is empty"<<endl;
                }
                else
                {
cout<<"List is not empty"<<endl;
                }
                break;

        case 11:
                cout<<"Enter position to get book ID: ";
                cin>>position;
                book_id=List.get(position);
                if(book_id!=-1)
```

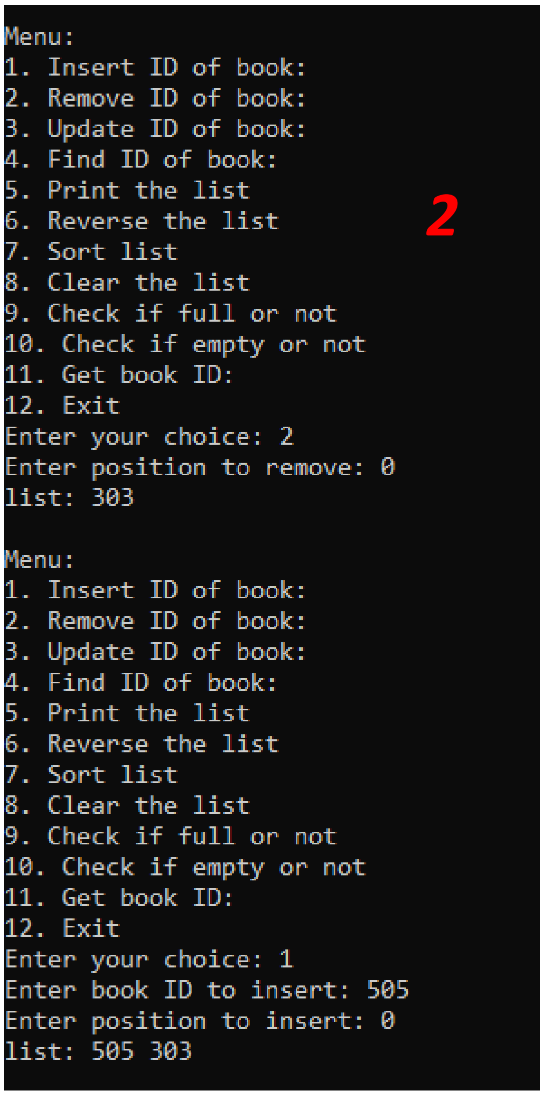
```
                        {
                                cout<<"Book ID at position "<<position<<" is "<<book_id<<endl;
                        }
                        break;

                case 12:
                        cout<<"Exiting program!"<<endl;
                        break;

                default:
                        cout<<"invalid choice, Please try again"<<endl;
                }
        }
        while(choice!=12);
   return 0;
}
```

Output-01:

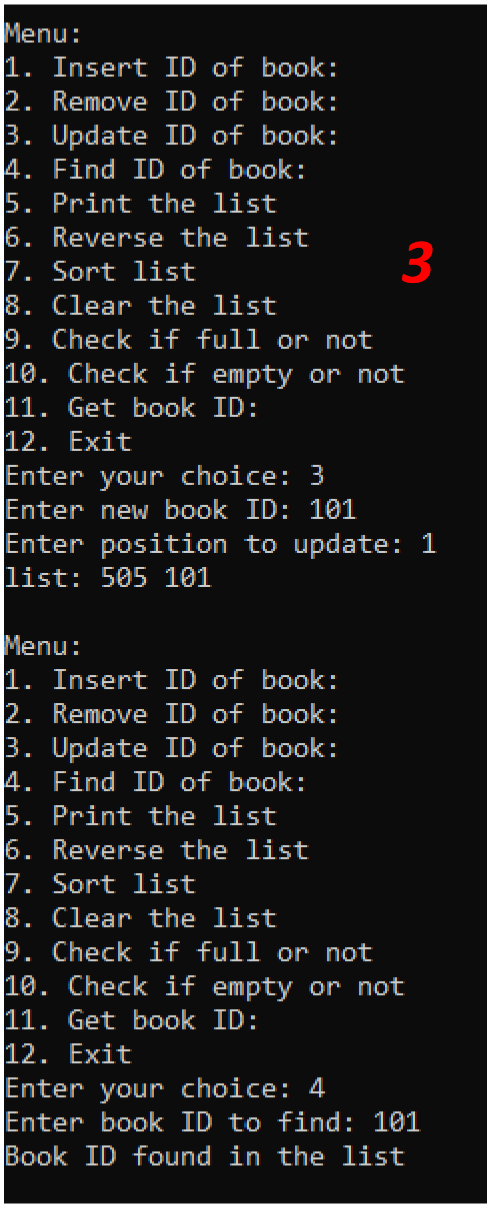
SEE BELOW



```
D:\SUMMER' 24\Data Structures LAB\LAB TA
Enter size of list: 2

Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 1
Enter book ID to insert: 404
Enter position to insert: 0
list: 404

Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 1
Enter book ID to insert: 303
Enter position to insert: 1
list: 404 303
```

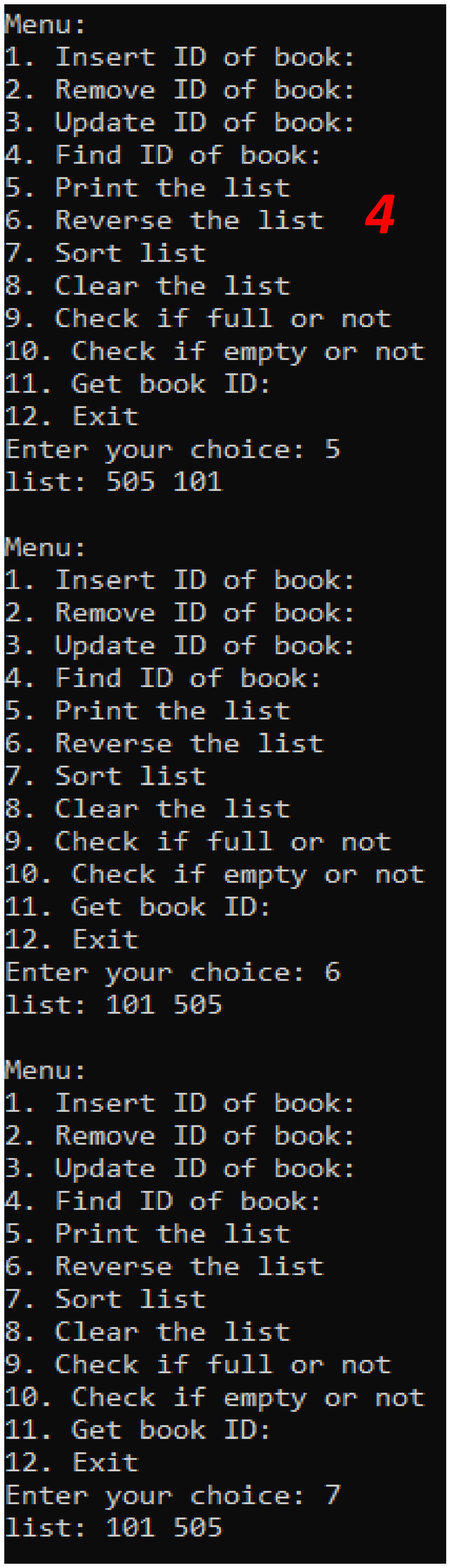


```
Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 2
Enter position to remove: 0
list: 303

Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 1
Enter book ID to insert: 505
Enter position to insert: 0
list: 505 303
```

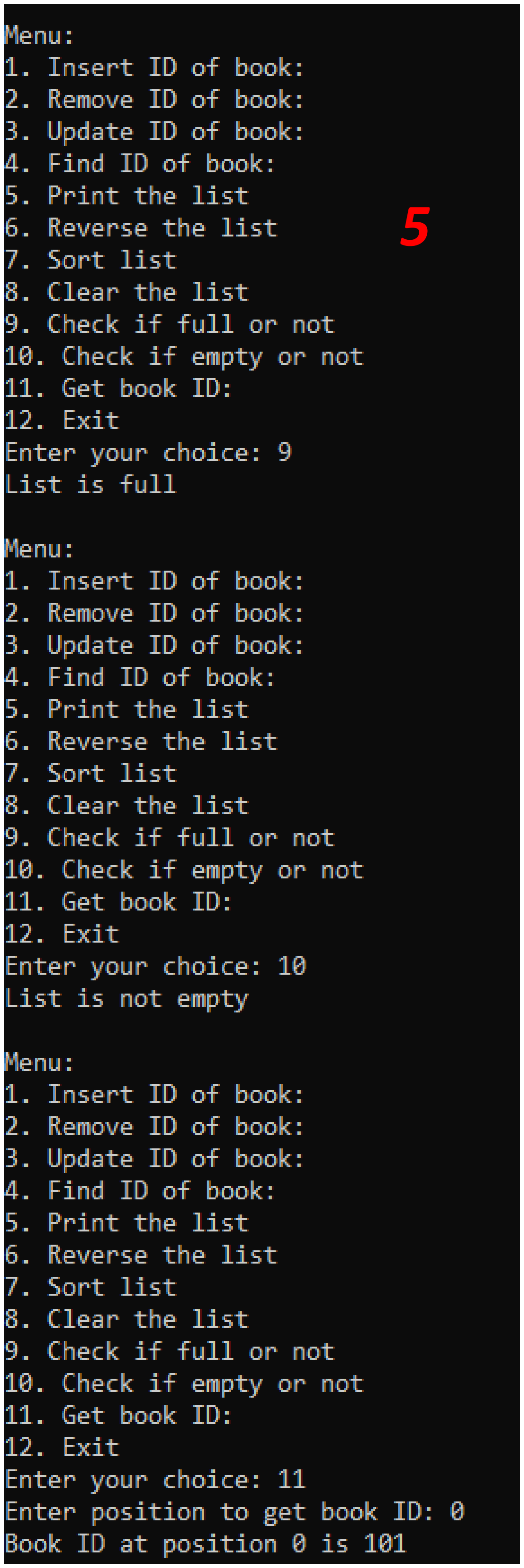


```
Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 3
Enter new book ID: 101
Enter position to update: 1
list: 505 101

Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 4
Enter book ID to find: 101
Book ID found in the list
```

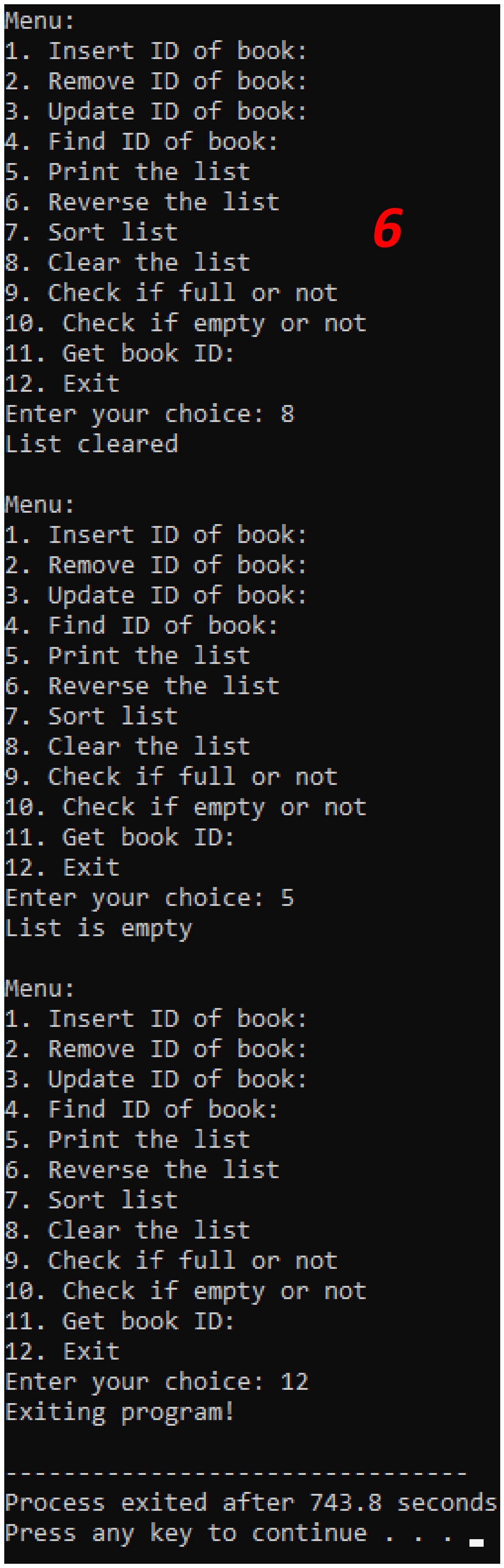
FOR MORE OUTPUTS SEE BELOW



```
Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 5
list: 505 101

Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 6
list: 101 505

Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 7
list: 101 505
```



```
Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 9
List is full

Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 10
List is not empty

Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 11
Enter position to get book ID: 0
Book ID at position 0 is 101
```



```
Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 8
List cleared

Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 5
List is empty

Menu:
1. Insert ID of book:
2. Remove ID of book:
3. Update ID of book:
4. Find ID of book:
5. Print the list
6. Reverse the list
7. Sort list
8. Clear the list
9. Check if full or not
10. Check if empty or not
11. Get book ID:
12. Exit
Enter your choice: 12
Exiting program!

--------------------------------
Process exited after 743.8 seconds
Press any key to continue . . .
```