Programa referente à Questão 4:

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

class Node{ //declaração da classe do nó

	public:

		Node(int el=0,Node *ptr=0)
		{
			info=el;
			next=ptr;
		}

		int info;
		Node *next;
};

class Stack{ //declaração da classe de pilha

	public:

		Stack() // construtor
		{
			head=0;
		}

		void push(int el) //função para adicionar um elemento à pilha
		{
			head=new Node(el, head);
```

```
	}

	void pop() //função para excluir um elemento da pilha
	{
		if(head==0)
{
			cout<<"\nPilha vazia"<<endl;
		}
		else if(head->next==0)
		{
			head=0;
		}
		else
{
			Node *tmp=head->next;
			delete head;
			head=tmp;
		}
	}

	int popEl() //função para excluir um elemento especifico
	{
		if(head == 0)
{
			cout<<"\nPilha vazia"<<endl;
			return -1;
		}
		else
		{
			return head->info;
		}
```

```
        }
        void clear() //função para excluir a pilha
        {
                Node *tmp=head;

                while(tmp!=0)
    {
       tmp=tmp->next;
                        delete head;
                        head=tmp;
                }
        }

        bool is_empty() //função para determinar se a pilha está vazia
        {
                return(head==0);
        }

        void printStack() //função para imprimir a pilha
        {
                for(Node *tmp=head;tmp!=0;tmp=tmp->next)
    {
                        cout<<"\nElemento:"<<tmp->info<<endl;
                }
        }

   private:
      Node *head;
};

class Queue //declaração da classe de fila
```

```
{
        public:
                Queue() //construtor da fila
                {
                        head=tail=0;
                }

                void enqueue(int el) //função para enfileirar elementos
                {
                        if(head == 0)
        {
                                head=tail=new Node(el, 0);
                        }
                        else
                        {
                                tail->next=new Node(el, 0);
                                tail=tail->next;
                        }
                }

                void dequeue() //função para desenfilar elementos
                {
                        if(!is_empty())
          {
                                cout<<"\nElemento removido: "<<firstEl();
                                Node *tmp=head;

                                if(head==tail)
          {
                                        head=tail=0;
                                }
```

```
                else
                {
                        head=head->next;
                }

                delete tmp;
        }
        else
    {
                cout<<"\nLista vazia, nao pode ser retirado nenhum
elemento";
        }
}

int firstEl() //função para retornar o primeiro elemento da fila
{
        if(head==0)
        {
                cout<<"\nFila vazia"<<endl;
                return -1;
        }
        else
        {
                return head->info;
        }
}

void clear_queue() //função para limpar a fila
{
        Node *tmp = head;
```

```
                while(tmp!= 0)
{
                        tmp=tmp->next;
                        delete head;
                        head=tmp;
                }
        }

        bool is_empty() //função para determinar se a fila está vazia
        {
                return(head==0);
        }

        void print_queue() //função para imprimir a fila
        {
                for(Node *tmp=head;tmp!=0;tmp=tmp->next)
{
                        cout<<"\nElemento:"<<tmp->info<<endl;
                }
        }

        void reverse_queue() //função para reverter a fila
        {
                Stack p;

                for(Node *tmp=head;tmp!= 0;tmp=tmp->next)
{
                        p.push(tmp->info);
                }

                clear_queue();
```

```
                        while(!p.is_empty())
        {
                                enqueue(p.popEl());
                                p.pop();
                        }

                }
   private:
      Node *head, *tail;
};

int main()
{

        Queue q; //criação de uma fila

        q.enqueue(25);
        q.enqueue(05);
        q.enqueue(2004);
        q.enqueue(30);
        q.enqueue(03);
        q.enqueue(1974);

        cout<<"\nFila:"<<endl;
        q.print_queue();

        q.reverse_queue(); //momento da inversão da fila

        cout<<"\nFila invertida:"<<endl;
        q.print_queue();
```

```
        return 0;
}

//fim do programa
```