Data Structures

Lab 4

#1

Implement stack using array. If the stack is empty print empty in pop, print and reverse.

The program should have:

1. Push(a)
2. Pop(d)
3. Print(p) – print in normal order(FIFO)
4. Reverse(r) – print in reverse order(LIFO)
5. Length(l)
6. Exit(x)

#1

Input:

```
a 1
a 2
a 3
a 4
a 5
p
r
l
d
d
d
d
d
d
p
r
x
```

Output:

```
1 2 3 4 5
5 4 3 2 1
5
empty
empty
empty
```

#2

Implement stack using linked list. The program has the same specifications as in the previous exercise.

#3

Make a program that will output binary representation of an integer using stack. Do so until a negative number is met.

Note: Input will allways be nonnegative.

Input:	Output:
5	101
10	1010
0	0

#4

Make a program that will check if brackets are balanced(closed) in a string until x is met.

Input:	Output:
{}	1
[]}	0
[{()}]	1
[}	0
([)]	0
x	

#5

Make a program that will simulate the forward and backward functions in a browser(the arrows that let you go through History) using stack. After every operation output current page. If movement is impossible print “invalid”. New page will erase the forward stack.

It should have:

1. New page(a) – takes a string
2. Forward(f)
3. Backward(b)
4. Exit(x)

#5

Input:

```
a google
f
a youtube
a weather
b
b
a wikipedia
f
x
```

Output:

```
google
invalid
youtube
weather
youtube
google
wikipedia
invalid
```