```c
/* LinkStackTest.c */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "LinkStack.h"

static void StringFree(void *elemAddr)
{
    free(*(char **)elemAddr);
}

int main(void)
{
    STACK intStack;
    StackNew(&intStack, sizeof(int), NULL);
    int i = 0;
    for (; i < 10; i ++)
    {
        if (0 == StackPush(&intStack, &i))
        {
            printf("intStack push key %d success\n", i);
        }
        else
        {
            printf("intStack push key %d fail\n", i);
        }
        if (!StackEmpty(&intStack))
        {
            int intTop;
            if (0 == StackTop(&intStack, &intTop))
            {
                printf("top of intStack is %d\n", intTop);
            }
            else
            {
                printf("get top element of intStack fail\n");
            }
        }
    }
    if (!StackEmpty(&intStack))
    {
        printf("size of intStack is %d\n", StackSize(&intStack));
    }

    for(; i >= 0; i--)
    {
        int intPop;
        if (0 == StackPop(&intStack, &intPop))
        {
            printf("data %d pop from intStack\n", intPop);
        }
        else
        {
            printf("pop from intStack fail\n");
        }

    }
    StackDispose(&intStack);

    printf("\n\n\n");

    STACK stringStack;
    StackNew(&stringStack, sizeof(char *), StringFree);
    char *name1 = strdup("jerry");
    char *name2 = strdup("pc");
    char *name3 = strdup("pcwl513");
    char *name4 = strdup("pcpc");

    char *strTop = NULL;
    StackPush(&stringStack, &name1);
    if (0 == StackTop(&stringStack, &strTop))
    {
        printf("top of stringStack is %s\n", strTop);
    }
    else
    {
        printf("get top element of stringStack fail\n");
    }
    StackPush(&stringStack, &name2);
    StackPush(&stringStack, &name3);
    StackPush(&stringStack, &name4);
    if (0 == StackTop(&stringStack, &strTop))
    {
        printf("top of stringStack is %s\n", strTop);
    }
    else
    {
        printf("get top element of stringStack fail\n");
    }
    if (!StackEmpty(&stringStack))
    {
        printf("size of stringStack is %d\n", StackSize(&stringStack));
    }

    i = 5;
    for(; i >= 0; i--)
    {
        char *strPop = NULL;
        if (0 == StackPop(&stringStack, &strPop))
        {
            printf("data %s pop from stringStack\n", strPop);
        }
        else
        {
            printf("pop from stringStack fail\n");
        }

    }
    StackDispose(&stringStack);
    return 0;
}
```