

## 数据结构教程第十七课实验三：栈的表示与实现及栈的应用

PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/138/2021\\_2022\\_\\_E6\\_95\\_B0\\_E6\\_8D\\_AE\\_E7\\_BB\\_93\\_E6\\_c98\\_138099.htm](https://www.100test.com/kao_ti2020/138/2021_2022__E6_95_B0_E6_8D_AE_E7_BB_93_E6_c98_138099.htm)

教学目的：掌握栈的

存储表示方式和栈基本操作的实现方法

教学重点：栈的基本

操作实现方法，栈的应用

教学难点：栈的存储表示

实验内容

：一、栈的实现

实现栈的顺序存储。

栈实现示例

```
#include #include #define ERROR 0 #define TRUE 1 #define FALSE
```

```
0 #define OK 1 #define EQUAL 1 #define OVERFLOW -1 #define
```

```
STACK_INIT_SIZE 100 #define STACKINCREMENT 10 typedef
```

```
int Status . struct STU{ char name[20]. char stuno[10]. int age. int
```

```
score. }. typedef struct STU SElemType. struct STACK { SElemType
```

```
*base. SElemType *top. int stacksize. }. typedef struct STACK
```

```
SqStack. typedef struct STACK *pSqstack. Status InitStack(SqStack
```

```
**S). Status DestroyStack(SqStack *S). Status ClearStack(SqStack
```

```
*S). Status StackEmpty(SqStack S). int StackLength(SqStack S).
```

```
Status GetTop(SqStack S,SElemType *e). Status Push(SqStack
```

```
*S,SElemType e). Status Pop(SqStack *S,SElemType *e). Status
```

```
StackTraverse(SqStack S,Status (*visit)()). Status InitStack(SqStack
```

```
**S) { (*S)=(SqStack *) malloc(sizeof(SqStack)).
```

```
(*S)->base=(SElemType *)malloc(STACK_INIT_SIZE
```

```
*sizeof(SElemType)). if(!(*S)->base)exit(OVERFLOW).
```

```
(*S)->top=(*S)->base. (*S)->stacksize=STACK_INIT_SIZE.
```

```
return OK. } Status DestroyStack(SqStack *S) { free(S->base).
```

```
free(S). } Status ClearStack(SqStack *S) { S->top=S->base. } Status
```

```
StackEmpty(SqStack S) { if(S.top==S.base) return TRUE. else return
```

```
FALSE. } int StackLength(SqStack S) { int i. SElemType *p. i=0.  
p=S.top. while(p!=S.base) {p . i . } } 100Test 下载频道开通，各类  
考试题目直接下载。详细请访问 www.100test.com
```