

**Problem 2 (5 points)**

For this problem, you will write the C functions to push a single integer onto a stack. The stack is stored in the global array **Stack**, and the variable **SP** is the stack pointer. Your function should return the value 1 if the item is successfully pushed onto the stack. If the item is not successfully pushed onto the stack, your function should return 0 and not alter the stack.

```
#define StackSize 1000
int Stack[StackSize];
int SP = 0;

int PUSH(int item)
{
```

```
}
```

**Problem 3 (8 points)**

The C code below generates a run-time error. Explain the cause of this error and how it could be fixed.

```
char alphabet[26];
main()
{
    char *str;
    int i = 0;

    for (c = 'a'; c <= 'z'; c++)
        alphabet[i++] = c;

    for (i = 0; i < 26; i++,str++)
        *str = alphabet[i];
}
```

**Problem 4** (*7 points*)

Write the function `strlen` that returns the length of a NULL-terminated string.

```
int strlen(char *str)
{
```

```
}
```

**Problem 5** (*5 points*)

The function below should return the value 1 if the global array `List` is sorted, and 0 otherwise. There is a bug in this function. Explain the bug, and describe what can be done to fix the problem. Assume that there are no duplicate entries in the array.

```
#define N 100
int List[N];

int is_List_sorted()
{
    int i;

    for(i = 0; i < N; i++)
        if (List[i] > List[i+1]) return 0;
    return 1;
}
```