

Suzanne Hill

However, I believe the error is in fill(arr) due to ^{calling} ~~using~~ a function without the operand &arr. The function called is a pointer variable.

```
array[0]=10;
array[1]=100;
arr=array;
}
int main(){
    int *arr=NULL;
    fill(arr);
    printf("arr[0]=%d ,arr[1]=%d\n",arr[0],arr[1]);
    return 0;
}
```

- 7) What is the output of the following program? Describe your understanding of the program and explain why it prints out the output.

```
#include <stdio.h>
int main(){
    char name[]="AliceBob";
    char *ptr=name;
    ① *ptr=*ptr+4;
    ② ptr=ptr+2;
    printf("%c",*ptr);
    ptr--;
    printf("%c",*ptr);
    ptr=name;
    printf("%c",*ptr);
    return 0;
}
```

① dereference the character A to 4 character over which is e.
 ② dereference the character A to 4 character over which is e.
 char *ptr=name; → points to first element in name
 *ptr=*ptr+4; → dereferences and moves the first char of name to the char e.
 ptr=ptr+2; → because e is the 4 character over from the first character in the array.
 printf("%c",*ptr); → pointer ptr2 moves 2 characters over which is i.
 ptr--; → Prints the char i.
 ptr--; → moves pointer ptr the 1st character over to the left
 printf("%c",*ptr); → prints the character e.
 ptr=name; → moves the ptr to the first character of the array name.
 printf("%c",*ptr); → Prints the char e from the dereference of #②.

Thus is why it will print out ile

- 8) Use gdb to debug the program below. Set a breakpoint at 'fact' function and examine the stack using "backtrace" to find the bug. Describe the bug you find.

```
#include <stdio.h>
int fact(int x){
    return x*fact(x-1);
}
int main(){
    printf("%d\n",fact(5));
}
```

Unable to use gdb keep on getting not in executable format; file format not recognized.