Suzanne. Hill

However, I believe the error is on fill(arr) due to ~~using~~ calling a function without the ~~apersand~~ &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.

① ② ~~defeference~~ dereference the character A to 4 character over which is e.

```
#include<stdio.h>
int main(){
   char name[]="AliceBob";
③ char *ptr=name;  — points to first ~~element~~ char in name
② *ptr=*ptr+4; 5 — dereferences and ~~moves~~ changes the first char of name to the char e (because e is the 4 character over from the first character in the Array.)
   ptr=ptr+2;  → pointer ptr moves 2 characters over which is i.
   printf("%c",*ptr);  → Prints the char. i.
   ptr--;  → moves pointer ptr 1 character ~~over~~ to the left
   printf("%c",*ptr);  → prints the character l.
   ptr=name;  → moves the ptr to the first character of the Array name.
   printf("%c",*ptr);
   return 0;}  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.

Unable to use gdb keep on getting not in executable format: File format not recognized.

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