```c
char* partialCopyString(char* str, bool copy_even, int* out_len) {
//BAD CODING : not initiallize to NULL
        char* OUT;
    if (copy_even) {
//CORECTNESS : if we allocate memory with a valid pointer we loose it
        out_len = malloc(sizeof(int));
//CORECTNESS : if strlen(str) == 7 for example out_len will recive 3 and not 4
        *out_len = strlen(str) / 2;
//BAD CODING : in all operating systems char = 1 bype but still i would write it
//BAD CODING : needs to check if memory allocation failed
//CORECTNESS : for new string we need to allocate memory of [strlen(str) + 1]
    OUT = malloc(*out_len);
//CORECTNESS : we will read from a non allocate memory in last iteration
    for (int i = 0; i < strlen(str); i += 2) {
        OUT[i / 2] = str[i + 1];
    }
    }else{
//CORECTNESS : if we allocate memory with a valid pointer we loose it
        out_len = malloc(sizeof(int));
        *out_len = strlen(str) / 2 + strlen(str) % 2;
//BAD CODING : in all operating systems char = 1 bype but still i would write it
//BAD CODING : needs to check if memory allocation failed
//CORECTNESS : for new string we need to allocate memory of [strlen(str) + 1]
    OUT = malloc(*out_len);
    for (int i = 0; i < strlen(str); i += 2)  {
        OUT[i / 2] = str[i];
    }
    }
    return OUT;
}
```