Part 1

1. Write a function about string copy, the strcpy prototype "char* strcpy (char* strDest, const char* strSrc);". Here strDest is destination string, strSrc is source string.
    a. Write the function strcpy, don't call the C string library.

```c
#include <stdio.h>
char* strcpy(char* strDest, const char* strSrc){
int i;
char len;

for(len=0; strSrc[len]!='\0'; ++len);
for(i=0; i<len; ++i){
strDest[i] = strSrc[i];
}
return strDest;
}

int main(){
char source[1000];
char dest[1000];
printf("Enter Source string: ");
scanf("%[^\n]", &source);
strcpy(dest,source);
printf("The destination string is: %s\n", dest);
return 0;
}
```

```
[ttahir1@gsuad.gsu.edu@snowball Lab10]$ vi strcpy.c
[ttahir1@gsuad.gsu.edu@snowball Lab10]$ [ttahir1@gsuad.gsu.edu@snowball Lab10]$
[ttahir1@gsuad.gsu.edu@snowball Lab10]$ gcc -o strcpy strcpy.c
[ttahir1@gsuad.gsu.edu@snowball Lab10]$ ./strcpy
Enter Source string: This is the source
The destination string is: This is the source
```

    b. Here strcpy can copy strSrc to strDest, but why we use char* as the return value of strcpy?
        i. We use char* as the return type because strDest is being returned and it is initialized as a char, also what is being copied is strSrc which is a char, so strDest has to be initialized as a char and returned as a char.

Part 2

1. Write a program findStr.c that finds the "smallest" and "largest" in a series of words. After the user enters the words, the program will determine which words would come first and last if the words were listed in dictionary order. The program must stop accepting input when the user enters a four-letter word. Assume that no word is more than 20 letters long. An interactive session with the program might look like this:
    a. Attach the source code of your C program into the answer sheet.

```c
#include <stdio.h>
#include <string.h>

int main(){
char smallest_word[21]="";
char largest_word[21]="";
char word[21];
printf("Enter word: ");
scanf("%s",word);

strcpy(smallest_word,word);

while(strlen(word)!=4){

if(strcmp(word,largest_word)>0){
strcpy(largest_word,word);
}

if(strcmp(word,smallest_word)<0){
strcpy(smallest_word,word);
}

printf("Enter word: ");
scanf("%s",word);

}
printf("Smallest word: %s\n", smallest_word);
printf("Largest word: %s\n", largest_word);
return 0;
}
```

    b. Run the C program, attach a screenshot of the output in the answer sheet.

```
[ttahir1@gsuad.gsu.edu@snowball Lab10]$ vi findStr.c
[ttahir1@gsuad.gsu.edu@snowball Lab10]$ [ttahir1@gsuad.gsu.edu@snowball Lab10]$
[ttahir1@gsuad.gsu.edu@snowball Lab10]$ gcc -o findStr findStr.c
[ttahir1@gsuad.gsu.edu@snowball Lab10]$ ./findStr
Enter word: dog
Enter word: zebra
Enter word: rabbit
Enter word: catfish
Enter word: walrus
Enter word: cat
Enter word: fish
Smallest word: cat
Largest word: zebra
```

```
#include <stdio.h>
#include <string.h>

int main(){
	char smallest_word[21]="";
	char largest_word[21]="";
	char word[21];
	printf("Enter word: ");
	scanf("%s",word);

	strcpy(smallest_word,word);
	while(strlen(word)!=4){
		if(strcmp(word,largest_word)>0){
			strcpy(largest_word,word);
			}
		if(strcmp(word,smallest_word)<0){
			strcpy(smallest_word,word);
			}
		printf("Enter word: ");
		scanf("%s",word);
	}
	printf("Smallest word: %s\n", smallest_word);
	printf("Largest word: %s\n", largest_word);
	return 0;
	}
```