02 Strings and Enums

Strings in C

- as a pointer to char

```
char *p = "abcde";
```

- as an array of char

```
char s[] = "abcde";
```

```
print("%d %d", sizeof(p), sizeof(s));
```

String Function Implementation

```
#include "stdio.h"

int strlenB(char* s) {
    int i = 0;
    while (s[i] != '\0') {
        i++;
    }
    return i;
}

char* strcpyB(char *s, char *d) {
    int len_s = strlenB(s);
    for(int i = 0; i <= len_s;i++) {
        d[i] = s[i];
    }
    return d;
}

char* str_rev(char *s, char *d) {
    int len_s = strlenB(s);
```

```
    for(int i = 0; i< len_s; i++) {
        d[i] = s[len_s - 1 - i];
    }
    d[len_s] = '\0';
    return d;
}

int main() {
    char *p = "xyz\0dsalkfjds";
    char s[] = "abcde\0kjdakfjsh"; // abcde\0

    // printf("%d %d\n",sizeof(p),sizeof(s));
    // printf("%s %s\n",p, s);
    // printf("%d %d\n", strlen(p), strlen(s));
    // strcpy(s,p);
    printf("%s\n",str_rev(p, s));

    return 1;
}
```

String Functions

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

- `int strlen(char *s)` : returns the length of the string pointed by s (ie lenth upto the first `\0` in memory).
- `char* strcat(char* s1, char* s2)` : concatenates s1 with s2, stores it in s1 and returns s1.
- `int strcmp(char *s1,char *s2)` : returns negative int if s1 is lex. smaller than s2, returns 0 if equal, returns positive int if s1 is lex. greater than s2.
- `char* strcpy(char* s1, char* s2)` : copies s2 in to s1 and returns s1.

Social Network

Name	Age	Rel Status	Friends
Alice	24	Single	Diestel, Eve
Bob	28	Maried	Alice
Charlie	20	Single	Diestel
Diestel	27	Not Mentioned	Alice, Eve, Charlie
Eve	25	Engaged	Diestel, Alice

Define a Person (Profile)

```
struct Person {
    char name[100];
    int age;
    int rel_status;
};
```

Implemeting Rel Status as int, requires us to keep in mind the mapping between Single, Maried, Not Mentioned, Engaged and integers.

> ## Can we specify this in code??

Enums

```
typedef enum Weekday {
    Sunday,
    Monday,
    Tuesday,
    Wednesday,
    Thursday,
    Friday,
    Saturday
} Weekday;
```

```
Weekday today = Wednesday;
printf("Day %d",today+1);
printf("Size of enum variable = %d bytes",
        sizeof(today));
```

Enums : Changing default values

```
typedef enum Weekday {
    Sunday = 1,
    Monday,
    Tuesday,
    Wednesday,
    Thursday,
    Friday,
    Saturday
} Weekday;
```

```
Weekday today = Wednesday;
printf("Day %d",today+1);
```

Enums : interchangable with int

```
#include "stdio.h"

typedef enum Weekday {
    Sunday = 5,
    Monday = 3,
    Tuesday,
    Wednesday = 2,
    Thursday,
    Friday,
    Saturday
} Weekday;

int main() {
    Weekday today = Wednesday;;
    printf("Day %d\n",today+1);
    printf("Size of enum variable = %d bytes",
            sizeof(today));
    return 0 ;
}
```

Define a Person (Profile)

```
enum RelStatus {
    NotMentioned,
    Single,
    Engaged,
    Married
};

struct Person {
    char name[100];
    int age;
    enum RelStatus status;
};
```

Practice Problems

1. Implement `str_rev` in place. ie. it takes only one string (char *) as argument, reverses it.