Introduction to Programming

Labor 12

Important Dates

Endterm Test

- 2nd December 2022 (Friday)
- 2–4 pm, IK–201 **OR** 4–6 pm, IK–201

Retake Midterm Test

- 9th December 2022 (Friday)
- 2–4 pm, IK–201

Retake Endterm Test

- 9th December 2022 (Friday)
- 4–6 pm, IK–201

Exercise – Revision

1. Write a program which defines if in the input string there are two vowels next to each other.

   For example: people, tea

Solution

```
int i, flag=0;
char vowels[10]="aeiuo", s[50];
fgets(s, 50, stdin);

for (i=0;i<strlen(s)-1;i++)
        if (strchr(vowels,s[i]) && strchr(vowels,s[i+1]))
                flag=1;

if (flag==1)
        printf(”There is!\n");
else
        printf(”There isn’t!\n");
```

Conversion

#include <stdlib.h>

atof() Convert string to double.

- double x;
- char *stringptr;
- x = atof(stringptr);

atoi() Convert string to integer.

- int i;
- char *stringptr;
- i = atoi(stringptr);

atol() Convert string to long integer.

- long i;
- char *stringptr;
- i = atol(stringptr);

Exercise

- Input a string which contains numbers. We should define how many ’5’ characters it contains, after that define the sum of the digits of the number.

- For example:
  - Input: 525
  - Output:
    - Number of 5s: 2
    - Sum of the digits:12

Solution

```
int n, count = 0, i, sum = 0;
char s[20];
scanf("%s", s);
for (i = 0; i < strlen(s); i++)
        if (s[i] == '5')
            count++;
    printf(”Number of 5s=%d\n", count);
    n = atoi(s);
    while (n) {
        sum += n % 10;
        n /= 10;
    }
    printf(”Sum of the digits=%d\n", sum);
```

sscanf()

sscanf(s,”%d”,&n);

- reads formatted input from a string

```
char sentence [50]="Rudolph is 12 years old";
char s[20];
int i;
sscanf (sentence, "%s %*s %d", s, &i);
printf ("%s -> %d\n", s, i);
```

Output: Rudolph -> 12

Note: For scanf, the * indicates that the field is to be read but ignored.

sprintf()

sprintf(s,”%d”,n);

- write formated data to string

```
char s[50];
int a=5, b=3;
sprintf (s, "%d plus %d is %d", a, b, a+b);
puts(s);
```

Output:
5 plus 3 is 8

Exercise

- Declare a string which contains numbers, after calculate how many ’5’ characters the square of the number contains.
- Use the sscanf and sprintf functions!

Solution

```
int n, count = 0, i, nn;
char s[10] = "125", p[20];

sscanf(s, "%d", &n);
nn = n * n;
sprintf(p, "%d", nn);

for (i = 0; i < strlen(p); i++)
        if (p[i] == '5')
            count++;
 printf("p=%s count=%d\n", p, count);
```

File Handling

- A **file** represents a sequence of bytes on the disk where a group of related data is stored.
- File is created for permanent storage of data.
- It is a ready-made structure.

- In C language, we use a structure **pointer of file type** to declare a file.

```
FILE *file_pointer;
FILE *f;
```

Function Description

- fopen() create a new file or open an existing file
- fclose() closes a file
- fgetc() reads a character from a file
- fputc() writes a character to a file
- fscanf() reads a set of data from a file
- fprintf() writes a set of data to a file
- getw() reads an integer from a file
- putw() writes an integer to a file
- fseek() set the position to desire point
- ftell() gives current position in the file
- rewind() set the position to the beginning point

Opening or creating a file

f = fopen(filename, mode);

f = fopen(”input.txt”, ”r”);

- **Mode** can be of following types:
  - r — opens a text file in reading mode
  - w — opens or create a text file in writing mode.
  - a — opens a text file in append mode
  - r+ — opens a text file in both reading and writing mode
  - w+ — opens a text file in both reading and writing mode
  - a+ — opens a text file in both reading and writing mode
  - rb — opens a binary file in reading mode
  - wb — opens or create a binary file in writing mode
  - ab — opens a binary file in append mode
  - rb+ — opens a binary file in both reading and writing mode
  - wb+ — opens a binary file in both reading and writing mode
  - ab+ — opens a binary file in both reading and writing mode

Writing a file

- The file writing operations can be perfomed by the functions **fprintf** and **fputs** with similarities to read operations.

- The snippet for writing to a file is as:

```
FILE *f ;
f = fopen(“fileName.txt”, “w”);
fprintf(f, "%s %s %s %d", "We", "are", "in", 2022);
```

Reading from a file

- The file reading operations can be perfomed using functions **fscanf**, **fgets.**
- Both functions perform the same operations as that of printf and gets but with an additional parameter, the file pointer.
- So, it depends on you if you want to read the file line by line or character by character.

- And the code snippet for reading a file is as:

```
FILE *f;
f = fopen(“fileName.txt”, “r”);
fscanf(f, "%s %s %s %d", s1, s2, s3, &year);
```

Closing a file

- After every successful file operation, you must always close a file.
- For closing a file, you have to use **fclose** function.

- The snippet for closing a file is given as:

```
FILE *f ;
f= fopen(“fileName.txt”, “w”);
…
fclose(f)
```

Example

```
FILE *f, *g;
int a;
f = fopen("in.txt", "r");
g = fopen("out.txt", "w");
fscanf(f, "%d", &a);
fprintf(g, "Number: %d", a);
fclose(f);
fclose(g);
```

Exercise

- Write a program, which reads words with maximum 100 character length from the standard input until the ”***” and save the words into the out.txt file, which contains the ’a’ or ’b ’ characters.

Solution

```
FILE *f = fopen("output.txt", "w");
char s[100];
scanf("%s", s);
while (strcmp( s,"***"))
{
    if (strchr(s, 'a') || strchr(s, 'b'))
        fprintf(f, "%s\n", s);
    scanf("%s", s);
}
fclose(f);
```

Exercise

- Write a program which reads characters from the **text.txt** file until the end of the sentence (’.’) and shows in which positions it found an ’*’ character and how many ’*’ were there.
- Write the positions of the asterisks and their number into the **out.txt** and after list the content of the file.

text.txt

- The*weather****was*****not good, however** we****had*a nice***trip.

Solution

```
FILE *f, *g; char c;
int pos = 0, star = 0;

f = fopen("text.txt", "r");
g = fopen("out.txt", "w");

while ((c = fgetc(f)) != '.')
{
    pos++;
    if (c == '*')
        star++;
    else {
if (star) {
fprintf(g, "%d\t %d\n", pos-star,
star);
    star= 0;
        }
      }
   }

fclose(f);
fclose(g);
```

Solution

List the content of the **out.txt** file.

```
printf("out.txt content\n");

g = fopen("out.txt", "r");
while (fscanf(g, "%d\t%d\n",   &pos, &star) != EOF)
printf("%d\t%d\n", pos, star);

fclose(g);
```

Exercise

- Write a program, which read integer numbers from the **code.txt**, and which determines the character needed to be written from **textin.txt** into the **message.txt**.
- You have to restart counting from the printed character.

- code.txt
  - 2 3 3 2 4 3 3 1 4
- textin.txt
  - *swou%sc+czo+ef*ezpdezi&d/o

Solution

```
FILE *fcode, *ftextin, *fmessage;
int number;
char c;

fcode = fopen("code.txt", "r");
ftextin = fopen("textin.txt", "r");
fmessage=fopen("message.txt","w");

while(fscanf(fcode,"%d",&number)!=
EOF)
    {
        while (number--)
                c=fgetc(ftextin);
        fputc(c, fmessage);
    }
fclose(ftextin);
fclose(fcode);
fclose(fmessage);

fmessage=fopen("message.txt","r");

while((c=fgetc(fmessage))!=EOF)
        printf("%c",c);

fclose(fmessage);
```