

Introduction to Programming

Lab 3

Alaa Aldin Hajjar, Amer Al Badr, Damir Nurtdinov,
Ikechi Kalu Ndukwe, Marko Pezer

Agenda

- Reading and writing from/to files
- Loops and arrays exercises
- String exercises
- Pointer exercises

R/W Files

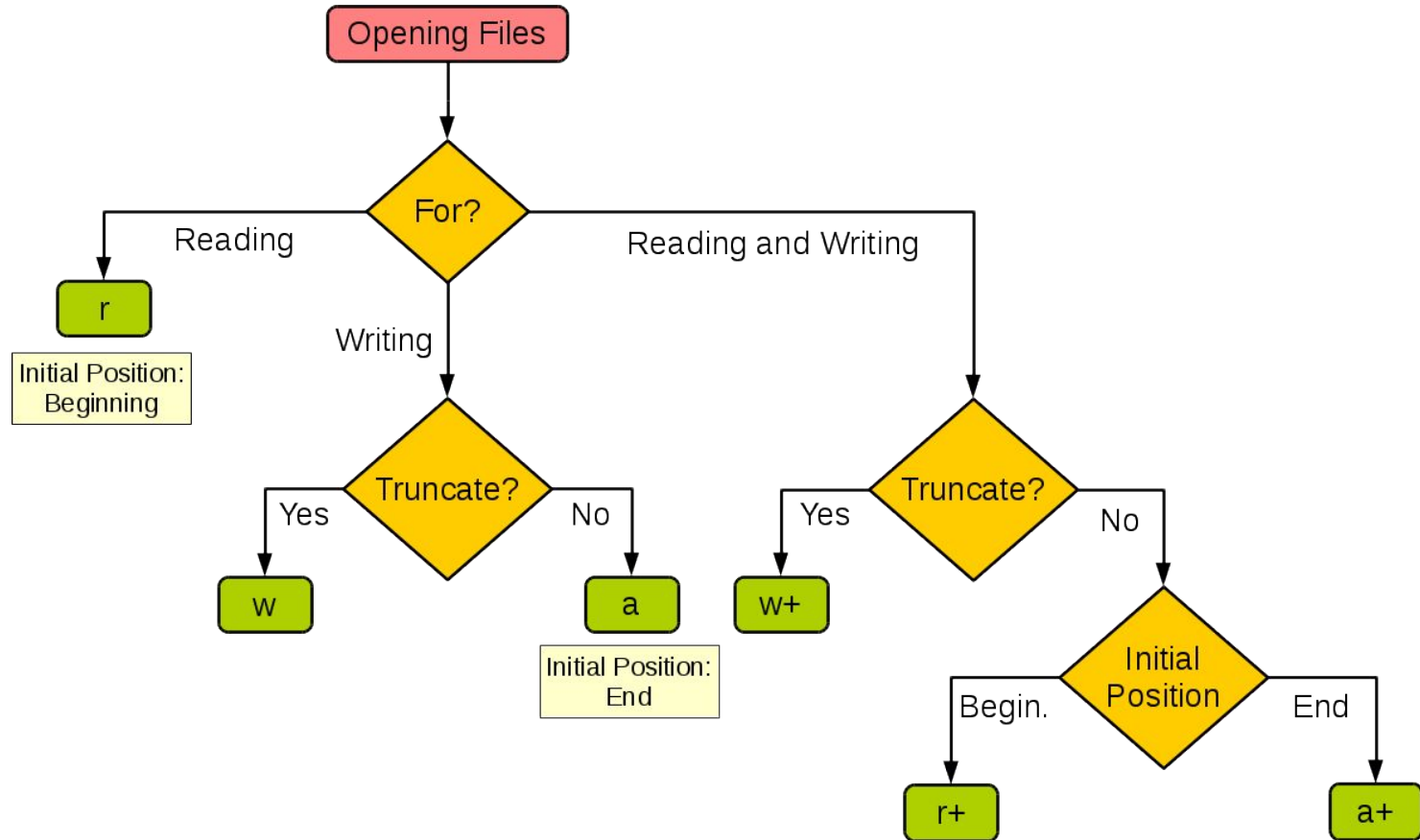
R/W files

fopen()

Declaration: FILE *fopen (const char *filename, const char *mode)

Modes:

- **r** – Opens a file in read mode and sets pointer to the first character in the file. It returns null if file does not exist.
- **w** – Opens a file in write mode. It returns null if file could not be opened. If file exists, data are overwritten.
- **a** – Opens a file in append mode. It returns null if file couldn't be opened.
- **r+** – Opens a file for read and write mode and sets pointer to the first character in the file.
- **w+** – opens a file for read and write mode and sets pointer to the first character in the file.
- **a+** – Open a text file in append mode for reading or updating at the end of the file. But, it can't modify existing contents.



R/W files

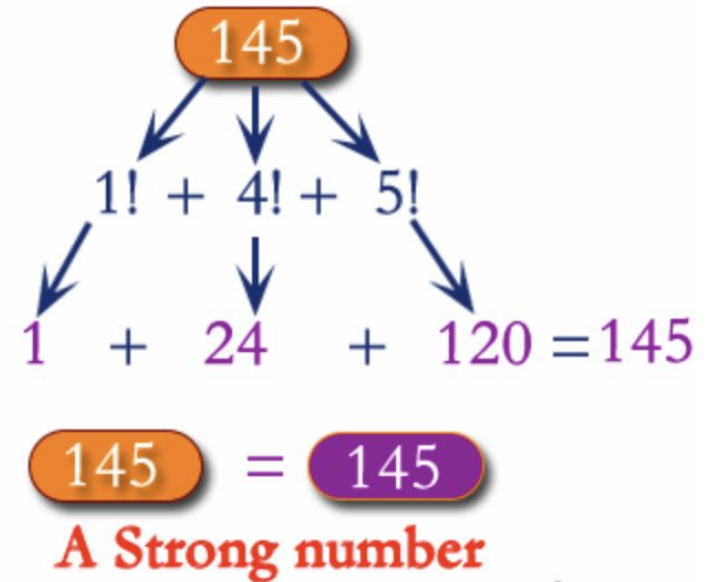
<code>fopen ()</code>	<code>fopen ()</code> function creates a new file or opens an existing file.
<code>fclose ()</code>	<code>fclose ()</code> function closes an opened file.
<code>gets ()</code>	<code>gets ()</code> function reads line from keyboard.
<code>fputs ()</code>	<code>fputs ()</code> function writes string to a file.
<code>fgets ()</code>	<code>fgets ()</code> function reads string from a file, one line at a time.
<code>fscanf ()</code>	<code>fscanf ()</code> function reads formatted data from a file.
<code>fprintf ()</code>	<code>fprintf ()</code> function writes formatted data to a file.

```
1  /* Open, write and close a file : */
2  # include <stdio.h>
3  # include <string.h>
4
5  int main( )
6  {
7      FILE *fp ;
8      char data[50];
9      // opening an existing file
10     printf( "Opening the file test.c in write mode" ) ;
11     fp = fopen("test.c", "w") ;
12     if ( fp == NULL )
13     {
14         printf( "Could not open file test.c" ) ;
15         return 1;
16     }
17     printf( "\n Enter some text from keyboard" \
18           " to write in the file test.c" ) ;
19     // getting input from user
20     while ( strlen ( gets( data ) ) > 0 )
21     {
22         // writing in the file
23         fputs(data, fp) ;
24         fputs("\n", fp) ;
25     }
26     // closing the file
27     printf("Closing the file test.c") ;
28     fclose(fp) ;
29     return 0;
30 }
```

Loops and Arrays

What is a strong number?

If the sum of factorial of the digits in any number is equal to the given number, then the number is called a STRONG number.



Exercise 1

Write a program to find Strong Numbers* within a range of numbers.
The program will receive 2 integers indicating the start and end of the range and calculates the strong numbers in the given range.

Input:

1

200

Output:

The strong numbers are: 1, 2, 145

ASCII (American Standard Code for Information Interchange)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Exercise 2

Write a program to print a histogram of the frequencies of different characters ordered by their frequency. The input is a string containing only lower case characters. (The horizontal histogram is a home exercise!)

Input: hello world

Output:

l ...
o ..
d .
e .
h .
r .
w .

l	o	d	e	h	r	w
.
.	.					
.						

Exercise 3

Write a program that will try to find a user password using bruteforce. User password can be at least 1 symbol and at most 3 symbols and contains only ASCII characters from 32 until 126.

Input:

u4!

Output:

found = u4!

number of attempts = ...

Pointers

Exercise 4

What will be the output of the programs?

Case A

```
1  #include <stdio.h>
2
3  void swap(int *ap, int *bp) {
4      int temp = *ap;
5      *ap = *bp;
6      *bp = temp;
7  }
8
9  int main() {
10     int a = 1, *ap = &a;
11     int b = 2, *bp = &b;
12
13     swap(ap, bp);
14     printf("%d%d%d%d%d\n",a,*ap,b,*bp);
15     return 0;
16 }
```

Case B

```
1  #include <stdio.h>
2
3  void swap(int *ap, int *bp) {
4      int *temp = ap;
5      ap = bp;
6      bp = temp;
7  }
8
9  int main() {
10     int a = 1, *ap = &a;
11     int b = 2, *bp = &b;
12
13     swap(ap, bp);
14     printf("%d%d%d%d%d\n",a,*ap,b,*bp);
15     return 0;
16 }
```

Case C

```
1  #include <stdio.h>
2
3  int main() {
4      int a = 1, *ap = &a;
5      int b = 2, *bp = &b;
6
7      int *temp = ap;
8      ap = bp;
9      bp = temp;
10
11     printf("%d%d%d%d%d\n",a,*ap,b,*bp);
12     return 0;
13 }
```

Exercise 5

What will be the output of the program?

```
1  #include <stdio.h>
2
3  int main() {
4      int array[] = {10, 20, 30};
5      int *pointer = array;
6
7      printf("%d\n", *pointer);
8      printf("%p\n", pointer);
9      printf("%d\n", *array);
10     printf("%p\n", array);
11
12     printf("%d\n", ++*pointer);
13     printf("%d\n", *++pointer);
14
15     int *pointer1 = array;
16     int *pointer2 = array;
17     printf("%d\n", *pointer1++ + ++*++pointer2);
18     return 0;
19 }
```

Exercise 6

Consider the following statements:

```
int *p;  
int i;  
int k;  
i = 42;  
k = i;  
p = &i;
```

After these statements, which of the following statements will change the value of *i* to 75?

- A) `k = 75;`
- B) `*k = 75;`
- C) `p = 75;`
- D) `*p = 75;`

Exercise 7

What will be the output of the program?

```
1  #include <stdio.h>
2  #include <string.h>
3  int main() {
4      char buf1[100] = "Hello";
5      char buf2[100] = "World";
6      char *ptr1 = buf1 + 2;
7      char *ptr2 = buf2 + 3;
8      strcpy(ptr1, buf2);
9      strcpy(ptr2, buf1);
10     printf("%s\n", buf1);
11     printf("%s\n", ptr1);
12     printf("%s\n", buf2);
13     printf("%s\n", ptr2);
14     return 0;
15 }
```

Exercise 8

Write a program to find length of a string using pointer. Do not use strlen().

Homework Exercises

Homework Exercise 1

Write a program to make such a pattern like a pyramid with numbers increased by 1.

Input: 4

Output:

```
  1
 23
456
78910
```

Homework Exercise 2

Write a program which deletes duplicate elements from array of integers.

The program will receive a number n ($0 < n < 1000$), followed by n elements of array where for each element e_i : $0 \leq e_i < 1000$.

Input:

13

0 1 1 3 1 1 4 2 5 4 0 0 5

Output:

0 1 3 4 2 5

Homework Exercise 3

Write a program to copy one string to another using pointer. Do not use strcpy().

Homework Exercise 4

Write a program to input and print elements of a two dimensional array using pointers and functions.

References

[Reading and writing from/on files.](#)