**Binary File Handling assignments**

Mandatory

1. WAP to

a. read 2 names (each of maximum length 20 char), and ages (uint) from the user.

b. Store the names in heap allocating required memory and ages in array. [You may also use a structure to keep name and age].

c. Write using fwrite() each name and age pair as comma separated pair to the binary file “out.bin” one per line . Close the file after saving.

d. Open the file, using fread() read each name and age pair and display

e. Verify if the displayed contents are same as that read from the user.

A screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generatedA black screen with white text

Description automatically generated

f. What was the

i. Size of Each name and age pair (based on input read)

Ans. Size = (Length of name + 1 for null terminator) + 1 (comma) + 4 bytes (age) + 1 (newline).

ii. Total sizeof 2 name and age pairs (based on input read)

Ans. Depends on the actual input names and ages.

iii. Size of file

Ans. Use ls -l out.bin to verify.

g. What do you observe between bytes read and bytes stored?

Ans . The bytes in the file correspond to the exact binary representation of the data written. Names are stored as strings, while ages are stored as unsigned integers.

h. Open file and see its content. Is it readable?

Ans. No, the file is in binary format and not directly human-readable.

i. Free allocated memory

Ans. Done in the main function using free().

2. WAP to read 2 or more server configuration details from the user and to process it as per given requirements. Refer the configuration structure as below.

#define MAX\_CFG 5 //Maximum number of configurations

#define MAX\_LEN 50

typedef struct config

{

char \*ipaddress; // pointer to a dotted ip address string

char name[MAX\_LEN];

unsigned short port;

}CONFIG;

CONFIG cfgarr\_in[MAX\_CFG];

CONFIG cfgarr\_out[MAX\_CFG];

a. Read and store the input configuration in cfgarr\_in

int read\_store\_cfg(cfgarr);

b. Display the stored configuration

void display(cfgarr, int count);

c. Store the configuration in a binary file “out.bin” using fwrite()

d. Read from the file using fread() and store in a cfgarr\_out

e. Display the contents of cfgarr\_out and verify it with content read from user in a).

A screenshot of a computer program

Description automatically generated

A screen shot of a computer screen

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated