Homework 7 Solutions – Binary Files

```
/*      Question 1
 *      Write a void function to copy a binary file into another file, 100 bytes at a time.
 *      The function receives the two FILE pointers (src and dest) as arguments.
 */

void copy (FILE *scr, FILE *dest)
{
        int     r;
        char    x[100];

        while ((r = fread (x, sizeof (char), 100, src)) > 0)
                fwrite (x, sizeof (char), r, dest);

        return;
}

/*      Question 2
 *      Write a function to initialize array x of size SIZE with integers read from a binary file.
 *      The function receives the name of the file as an argument.
 */
void init (char *name)
{
        FILE    *fp;
        int     i;

        fp = fopen (name, "br");
        if (fp == NULL)
        {
                printf ("no file\n");
                return;
        }

        if (fread (x, sizeof (int), SIZE, fp) < 1)
                printf ("error reading\n");

        fclose (fp);
        return;
}


/*      Question 3
 *      Write a function to return the number of even integers in a binary file.
 *      The function receives the file pointer (FILE *) as an argument.
 */
```

```
int even (FILE *fp)
{
        int     i, ret;
        int     counter = 0
        int     buffer[SIZE];

        if (fp == NULL)
                return -1;

        while ((ret = fread (buffer, sizeof(int), SIZE), fp)) > 0)
        {
                for (i = 0; i < ret; i++)
                        if (buffer[i] % 2 == 0)
                                counter++;
        }

        return (counter);
}

/*      Question 4
 *      Write a function to traverse a circular linked-list writing each node to a binary file.
 *      The function receives the file pointer (FILE *) as an argument.
 */
void binary_write (FILE *fp)
{
        NODE    *p;

        if (fp == NULL)
                return;

        if (head == NULL)
                return;

        p = head;
        do
        {
                fwrite (p, sizeof (NODE), 1, fp);
                p = p->next;
        }
        while (p != head);

        return;
}

/*      Question 5
 *      Write a function to create a linked list with nodes obtained from a binary file.
 *      The function receives the file pointer (FILE *) as an argument.
 */
```

```
void link (FILE *fp)
{
        NODE    *temp;
        int     counter = 0

        if (fp == NULL)
                return;

        while (1)
        {
                temp = (NODE *)malloc (sizeof (NODE));
                it (temp == NULL)
                        break;

                if (fread (temp, sizeof(NODE), 1), fp) < 1)
                {
                        free (temp);
                        break;
                }

                insert (temp);
        }

        return;
}

/*      Question 6
 *      Write a void function to compare two binary files.
 *      Your function should receive the file pointers (src and dest) and
 *      printf either “Files are equal.\n” or “Files are not equal.\n”.
 *      Use fread, fwrite, and an array to make the process more efficient!
 */
void compare (FILE *fp1, FILE *fp2)
{
        int     i;
        int     ret1;
        int     ret2;
        char    buffer1[SIZE];
        char    buffer2[SIZE];

        while (1)
        {
                ret1 = fread (buffer1, sizeof(char), SIZE, fp1);
                ret2 = fread (buffer2, sizeof(char), SIZE, fp2);

                //      not same size
                if (ret1 != ret2)
                {
```

```
            printf ("Files are not equal\n");
            return;
        }

        //      done
        if (ret1 < 1)
        {
            printf ("Files are equal\n");
            return;
        }

        for (i = 0; i < ret1; i++)
        {
            if (buffer1[i] != buffer2[i])
            {
                printf ("Files are not equal\n");
                return;
            }
        }
    }

    return;
}
```