

Apéndice C. Programa para obtener la matriz de evaluación

Correspondiente a la sección 2.3

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

#define TRUE 1
#define FALSE 0

int tabla[280][130];
int posiciones[129];
int alelos[278];
int clases[20][2];
double matrizreducida[21][129][8];
char combinaciones[100][20];
int combinacionactual=0;
int numtotal=1;

void lectura() {
    FILE *fp;
    int i, j;

    if ((fp = fopen("matriz.txt", "r+")) == NULL) {
        printf("error en la apertura del archivo!\n");
        exit(0);
    }

    for (j=0; j<129; j++) fscanf(fp, "%d", &posiciones[j]);

    for (i=0; i<278; i++) {
        fscanf(fp, "%d", &alelos[i]);
        for (j=0; j<129; j++) {
            fscanf(fp, "\t%c", &tabla[i][j]);
        }
    }
    fclose(fp);
}

void combinacionesaveinte() {
    char cadena[20];
    int posicion, columnas, longcad, i, k;
    int archivoabierto, maximo;
    FILE *fp;
    int numcombinaciones=0;
```

```

if ((fp=fopen("preguntas.txt", "w"))!=NULL) archivoabierto=TRUE;
for (posicion=0; posicion<=129; posicion++) {
    maximo=1;
    for (columnas=1; columnas<=10; columnas++) {
        if ((posicion+columnas) <= 129)
            if ((posiciones[posicion+columnas-1] - posiciones[posicion])<=19)
maximo=columnas;
    }
    columnas=maximo;
    if ((posicion+columnas) <= 129) {
        if (posiciones[posicion+columnas-1]-posiciones[posicion]<=19) {
            limpiacombinaciones();
            for (k=0; k<278; k++) { /* se recorren todos los alelos de la
tabla original */
                /* se obtiene una combinacion inicial */
                longcad=0;
                for (i=posicion; i<posicion+columnas; i++)
cadena[longcad++] = tabla[k][i];
                cadena[longcad] = '\0';
                if (combinacionNOchecada(cadena)) {
                    printf("%d : ", ++numcombinaciones);
                    insertacombinacion(cadena);
                    for (i=posicion; i<posicion+columnas; i++)
printf("%d,", posiciones[i]);

                    printf("\t%s\n", cadena);
                    if (archivoabierto) {
                        fprintf(fp, "%d : ", numcombinaciones);
                        for (i=posicion; i<posicion+columnas;
i++) fprintf(fp, "%d,", posiciones[i]);

                        fprintf(fp, "\t%s\n", cadena);
                    }
                }
            }
        }
    }
}
if (archivoabierto) fclose(fp);
}

void evaluapreguntas(int clase1, int clase2, char *cadena1, char *cadena2) {
    char cadena[20];
    int posicion, columnas, longcad, i, j, k;
    int archivoabierto, maximo;
    FILE *fp;
    int evaluacionpregunta1, evaluacionpregunta2;

```

```

if ((fp=fopen("evaluacion.txt", "a"))!=NULL) archivoabierto=TRUE;
printf("%d*%d:", clase1, clase2);
if (archivoabierto) fprintf(fp, "%d*%d:", clase1, clase2);

for (posicion=0; posicion<=129; posicion++) {
    maximo=1;
    for (columnas=1; columnas<=10; columnas++) {
        if ((posicion+columnas) <= 129)
            if ((posiciones[posicion+columnas-1] - posiciones[posicion])<=19)
maximo=columnas;
    }
    columnas=maximo;
    if ((posicion+columnas) <= 129) {
        if ((posiciones[posicion+columnas-1]-posiciones[posicion])<=19) {
            limpiacombinaciones();
            for (k=0; k<278; k++) { /* se recorren todos los alelos de la
tabla original */
                /* se obtiene una combinacion inicial */
                longcad=0;
                for (i=posicion; i<posicion+columnas; i++)
cadena[longcad++] = tabla[k][i];
                cadena[longcad] = '\0';
                if (combinacionNOchecada(cadena)) {
                    insertacombinacion(cadena);
                    evaluacionpregunta1=TRUE;
                    evaluacionpregunta2=TRUE;
                    for (i=posicion,j=0; i<posicion+columnas;
i++,j++) {
                        if (cadena[j] != cadena1[i])
evaluacionpregunta1=FALSE;
                    }
                    for (i=posicion,j=0; i<posicion+columnas;
i++,j++) {
                        if (cadena[j] != cadena2[i])
evaluacionpregunta2=FALSE;
                    }
                    if
((evaluacionpregunta1)||((evaluacionpregunta2)) printf("\t1");
                    else printf("\t0");
                    if (archivoabierto) {
                        if
((evaluacionpregunta1)||((evaluacionpregunta2)) fprintf(fp, "\t1");
                        else fprintf(fp, "\t0");
                    }
                }
            }
        }
    }
}

```

```

        }
    }
}
printf("\n");
if (archivoabierto) {
    fprintf(fp, "\n");
    fclose(fp);
}
}

```

```

void invoca_preguntas(FILE *fp, int archivoabierto, int comb, int clase1, int clase2, char
*cadena1, char *cadena2) {
    if (archivoabierto) {
        fprintf(fp, "%d: %s\n%s\n", comb, cadena1, cadena2);
    }
    //printf("%d: %d*%d: %s\n%s\n", comb, clase1, clase2, cadena1, cadena2);
    evaluapreguntas(clase1, clase2, cadena1, cadena2);
}

```

```

void main() {
    lectura();
    caracterizaclases();
    combinacionesaveinte();
    tabladebialelos();
}

```