```
2 #include <stdlib.h>
 3
 4
    struct NodoLista{
 5
        char dato;
 6
        struct NodoLista *ptrSiguiente;
 7
 8
 9
   void insertar(struct NodoLista **ptrS); ///**ptrSes.eb apuntador que representa *ptrSiguiente para poder
10 char eliminar (struct NodoLista **ptrS);
usarse en el programa//
11 void imprimeLista(struct NodoLista *ptrActual);
12
13 int main(){
 14 struct NodoLista *ptrInicial; 30 25bibieq and sup a nigrous spiral aid as onlive at become the
    ptrInicial = NULL;
 15
 16 char letra:
    int opcion=1;
 17
                                                                      Considero que [Printfles uno
    while (opcion != 0) {
 18
                                                                      ilamada al sistema porque
        printf("Ingresa la opcion deseada\n"
 19
                                                                      nos permite la impresión
        printf("1.- Ingresar\n2.- Sacar\n3.- imprimir\n0.- Salir\n"
 20
                                                                      de un dato deseado en el
         scanf ("%d", &opcion); sel requisione annual mentione
                                                                     monitor actual Prints
 21
 22
         fflush (stdin) ;
                                                                       contiene autorización
         switch (opcion)
 23
                                                                     del s.o. para tomar el mensaje
24
            case 1:
                                                                     y poder imprimirlo.
                insertar(&ptrInicial); //ESTOY ASIGNADO A LA FUNCIÓN
 25
                                                             SE USA EL DOBLE APUNTADOR EN EL PROTOTIPO DE LA
                         nento persistente durante la saguiente de
                                                                     Considero qui scanfi es una
 26
                break;
                                                                     ilamada al sistema ya que
 27
                                                                     se carga un determinado tipo de dato en una dirección
 28
            case 2:
                                                                     de memoria que puede reabir ese tipo de dato.
 29
                letra = eliminar(&ptrInicial);
 30
                printf("El valor eliminado es %c\n", letra)
 31
                break;
 32
                                                                      [fflush (stdin) es una función
 33
            case 3:
                imprimeLista(ptrInicial);
                                                                       que realiza la limpieza
 34
                break; Glos on obis man sendog sim
                                                                      del buffer de entrada al hacer esto libela memoria es
 35
 36
                                                                      por esto que la considero
             default:
 37
                                                                      una uamada al sistema.
 38
                break;
 39
 40
 41
     return 0;
 42
 43
 44
 45
 46
    void insertar(struct NodoLista **ptrS){
 47
         struct NodoLista *ptrActual; 0.21 v 002.4 onthe 4b abibing and a simble viupe obe soint
 48
         struct NodoLista *ptrAnterior; observation and a struct NodoLista *ptrAnterior;
 49
         struct NodoLista *ptrNuevo;
 50
         char letra
 51
         printf "Ingresa la letra deseada: \n"
 52
         scanf("%c", &letra);
 53
                                                                           matted le pide memoria
 54
         fflush (stdin);
                                                                            al sistema operativo para
 55
         ptrNuevo = (struct NodoLista *) malloc(sizeof(struct NodoLista));
                                                                            usarla dentro de mi
 56
         if (ptrNuevo != NULL) {
                                                                           proceso. Es por eso que
 57
             ptrNuevo->dato = letra;
                                                                            considero es una liama-
            printf("\n58ptrNuevo->dato: %c\n",ptrNuevo->dato
                                                                           da al sistema de tipo
 58
 59
             ptrAnterior = NULL;
                                                                            control de procesos
 60
             //printf("ptrActual %p\n",ptrActual);
             ptrActual = *ptrS:
 61
            printf("62*ptrS:%p\n", *ptrS)
 62
            printf("63ptrActual: %p *ptrS: %p\n",ptrActual, *ptrS)
```

1 #include <stdio.h>

```
64
 65
                       while (ptrActual != NULL && letra > ptrActual->dato) {
 66
                              printf("66%p\n",ptrAnterior)
 67
                               ptrAnterior = ptrActual;
                              printf("68 ptrActual: %p ptrActual->dato: %c\n" ptrActual ptrActual->dato);
 68
 69
                               ptrActual=ptrActual->ptrSiguiente;
 70
                               printf("70ptrActual: %p\n",ptrActual);
 71
                       una contracción en el ingreso de la mayoria de los trabajadores. Si bie) (LLUM == valentario) i
 72
                              ann los dátos sobre la recuperación siguen siendo may incipi
 73
                              printf("74*ptrs: %p\n", *ptrs);
 74
                                *ptrS=ptrNuevo;
 75
                               printf("76ptrNuevo: %p\n",ptrNuevo);
 76
 77
                       |else{
 78
                               ptrAnterior->ptrSiguiente = ptrNuevo; solloups no obirrupo ol no sissinò reneq sneq si visv
 79
                              printf("80ptrNuevo: %p\n",ptrNuevo);
 80
 81
                               ptrNuevo->ptrSiguiente = ptrActual;
                               Fstos sectores son el comercio al menud o, la hoct (latactual: %n/q% "n/q%" sectores son el comercio al menud o, la hoct (latactual: %n/q%" n/q%" n/q%
 82
 83
 84
 85
 86
                else
 87
                       printf("No hay espacio en memoria\n") de es de la producción de la producción de la producción de la printf("No hay espacio en memoria\n").
 88
 89
 90
 91
 92
 93
         char eliminar (struct NodoLista **ptrS) {
 94
                char letra;
                struct NodoLista *ptrActual; mehang al mog kohataga kan kanataga kal ma koshama ah majandintkid
 95
 96
                struct NodoLista *ptrAnterior;
 97
                struct NodoLista *ptrTemp;
 98
                printf("Que letra deseas eliminar?\n"
 99
                scanf(" %c", &letra);
100
101
102
                if (letra == (*ptrS)->dato) {
                                                                                                                                                   Ifree ( ) me permite liberar
103
                        ptrTemp = *ptrS
                                                                                                                                                    Ta memoria asignada
                       printf("104*ptrS:%p",*ptrS)
104
                        *ptrS = (*ptrS)->ptrSiguiente;
                                                                                                                                                    dinamicamente, es decir
105
                                                                                                                                                   en este caso usando Malloc
                        printf("106(*ptrS)->ptrSiguiente: %p\n", (*ptrS)->ptrSiguiente)
106
                                                                                                                                                   Al tener esta función
107
                       free (ptrTemp);
                                                                                                                                                    sabemos que es uno
                       return letra:
108
                                                                                                                                                    Hamada di sistema de
109
                                                                                                                                                     control de procesos porque
110
                ptrAnterior = *ptrS
                                                                                                                                                    se libera memoria.
                printf("111*ptrS:%p\n",*ptrS)
111
                ptrActual = (*ptrS)->ptrSiguiente;
112
                printf("113 (*ptrS)->ptrSiguiente: %p\n", (*ptrS)->ptrSiguiente)
113
114
115
                while (ptrActual != NULL && ptrActual -> dato != letra) {
116
                                                                                                                                  128 Threshe i
                       ptrAnterior = ptrActual;
117
                       printf("117ptrActual: %p\n".ptrActual)
                       ptrActual = ptrActual->ptrSiguiente;
118
119
                       printf("117ptrActual->ptrSiguiente: %p\n",ptrActual >ptrSiguiente)
120
121
                if (ptrActual != NULL) {
122
123
                       ptrTemp = ptrActual;
124
                       printf("ptrActual: %p\n",ptrActual)
125
                       ptrAnterior->ptrSiguiente = ptrActual->ptrSiguiente;
                        printf("ptrActual->ptrSiguiente: %p",ptrActual->ptrSiguiente
126
                       free (ptrTemp) ;
127
128
                        return letra:
129
```

```
return '\0';
130
131
132
133
     void imprimeLista(struct NodoLista *ptrActual){
         if (ptrActual==NULL) {
134
             printf("La lista esta vacia\n");
135
136
             return;
137
138
        printf("La lista es: \n");
139
         while (ptrActual != NULL) {
```

printf("%c ===>", ptrActual->dato)

printf("NULL\n"

140

141 142

143

144 145 146

147 148 recursos economicos de emergencia para los hogares.

[1.2] no recuperación de los empleos obicados a la mitad de la distribución de los empleos obicados a la mitad de la distribución de los empleos obicados a la mimero de empleos

pere pagados en les economias recresentan un patrón de recuperación

post panderna una cupanta. Por un lado, porque suede indicar el faicio

de un proceso de pelativación, en donde los empleos ubicados en la parte

muermedia de la distribución son remplazados por empleos en ambas o

en una sola de las colas de la distribución de ingreso laboral. Por otro, y

licado con el último enunciado, el que los sendens en la parte interior

de distribución se hayan acciperado más rápido sugiere que esta po-

entrada, es posible que su recaperación más acelerada se debe a que la

Vale la pena recerdar que la pandemia no neutrió en un vacío, sino que estaban ocurriendo.

procesos de transformación importantes al interior del morrado la boral. Desde la primera decada de siglo XXI se ha observado una caída en los rendimientos salariales de la educación superior, explicade en buena medida por el desajuste entre obrita y demanda de trabalo calificado o la mala calgnación de este último en la economía (Levy y Lópea Colva, 2019). Dieno desajuste ha estado compañado de una caída en el rendiciento de la experiencia laboral en el mercado (Campos Vázquez, Lópea-Caiva y Lostig. 2016). En su conjunto, estos dos patrones producion una reducción en la desigualdad del ingreso laboral en México. En este escenacio, una recuperación precarizante y polarizante puede revertir el cierre de la desigualdad en el ingreso laboral, a la par de implicar circi de la desigualdad en el ingreso laboral, a la par de implicar