```
// Ejercicio 1

#include <iostream>
#include <vector>
using namespace std;
struct Registro {
	int llave;
	string dato;
};

// Implementa bool insertar(int llave, const std::string&dato) que verifique 0 <
// ky < 0-9 COmprobar que la posicion este libre, si lo esta, que devuelva true,
// si no false Almacenamiento, que use un std::vector<Registro> de tam 10
// Inicializar todas las posiciones con llave =-1 para indicar vacio
// Busqueda, string buscar(int llave) que: compruebe que 0 < ky < 0-9
// Si tiene llave == llave, que retorne el dato, de lo contrario que imprima "No
// encontrado" Menu sencillo con 3 opciones

void Regigigas(std::vector<Registro>& reg) {
	reg.resize(10);  // Asegurar tamaño fijo de 10
	for (int i = 0; i < 10; ++i) {
		reg[i].llave = -1;  // Llave -1 indica posición vacía
		reg[i].dato = "";

  }
}

bool insertar(std::vector<Registro>& reg, int llave, const std::string& dato) {
	if (llave < 0 || llave >= 10) {
		std::cout << "Llave fuera del rango, debe de ser de 0 a 9.\n";
		return false;

  }
	if (reg[llave].llave == -1) {  // Comprobar si la posición está libre
		reg[llave].llave = llave;
		reg[llave].dato = dato;
		return true;

  }
  else {
		std::cout << "La posicion ya ha sido ocupada.\n";
		return false;

  }
}

std::string sears(const std::vector<Registro>& reg, int llave) {
	if (llave < 0 || llave >= 10) {
		std::cout << "Llave fuera del rango, debe de ser de 0 a 9.\n";
		return "";

  }
	if (reg[llave].llave == llave) {  // Verificar si la llave coincide
		return reg[llave].dato;

  }
  else {
		std::cout << "Llave vacia\n";
		return "";
```

```
  }
}

int main() {
      std::vector<Registro> reg;
      Regigigas(reg);

      int opcusr = 0;

      do {
            cout << "Bienvenido" << endl;
            cout << "Por favor, elija una opcion:" << endl;
            cout << "1) insertar registro" << endl;
            cout << "2) Buscar registro" << endl;
            cout << "3) Salir" << endl;
            cin >> opcusr;

            switch (opcusr) {
            case 1: {
                  int llave;
                  string dato;

            cout << "Ingrese la llave (Considere su rango de 0-9): ";
                  cin >> llave;
                  cout << "Ingrese el dato: ";
                  cin.ignore();
                  getline(cin, dato);

            if (insertar(reg, llave, dato)) {
                        cout << "Registro insertado correctamente.\n";

      }
                  break;

    }
            case 2: {
                  int llave;
                  cout << "Ingrese la llave a buscar: ";
                  cin >> llave;
                  string resultado = sears(reg, llave);
                  if (!resultado.empty()) {
                        cout << "Dato encontrado: " << resultado << "\n";

      }
                  break;

    }
            case 3:
                  cout << "Saliendo...\n";
                  break;
              default :
            cout << "Opción inválida.\n";

    }

  }
  while (opcusr != 3);
```

```
      return 0;
}

// Ejercicio 2//Ejercicio 2
#include <iostream>
#include <string>
#include <vector>

using namespace std;

struct Registro {
      int llave;
      string dato;
      Registro* next;


};

int hashFunction(int llave) {
    return llave % 5; }

bool insertar(vector<Registro*>& tabla, int llave, const string& dato) {
      int h = hashFunction(llave);
      Registro* nuevo = new Registro(llave, dato);

      if (tabla[h] == nullptr) {
            tabla[h] = nuevo;

  }
      return true;
}

string buscar(const vector<Registro*>& tabla, int llave) {
      int h = hashFunction(llave);
      Registro* actual = tabla[h];

      while (actual != nullptr) {
            if (actual->llave == llave) {
                  return actual->dato;

    }
            actual = actual->next;

  }
      return "No Encontrado";
}

int main() {
      vector<Registro*> tabla(5, nullptr);

      int opcusr;
      do {
            cout << "Menu:" << endl;
            cout << "1. Insertar registro" << endl;
            cout << "2. Buscar registro" << endl;
            cout << "3. Salir" << endl;
            cin >> opcusr;

            switch (opcusr) {
```

```
            case 1: {
                  int llave;
                  string dato;
                  cout << "Ingrese la llave: ";
                  cin >> llave;
                  cout << "Ingrese el dato: ";
                  cin.ignore();
                  getline(cin, dato);
                  if (insertar(tabla, llave, dato)) {
                        cout << "Registro insertado correctamente.\n";

      }
                  break;

    }
            case 2: {
                  int llave;
                  cout << "Ingrese la llave a buscar: ";
                  cin >> llave;
                  string resultado = buscar(tabla, llave);
                  cout << "Resultado: " << resultado << endl;
                  break;

    }
            case 3:
                  cout << "Saliendo...\n";
                  break;
              default :
            cout << "Opcion invalida.\n";

    }

  }
  while (opcusr != 3);

      for (int i = 0; i < 5; ++i) {
            Registro* actual = tabla[i];
            while (actual != nullptr) {
                  Registro* temp = actual;
                  actual = actual->next;
                  delete temp;

    }

  }

      return 0;
}
// ejercicio 3
// crea un programa que gestione un arreglo de enteros en memoria dinamica
// se debe usar new para asignar espacio segun el tamano que indique el usuario
// luego llenar el arreglo, mostrarlo y finalmente liberar la memoria con
// delete[]

// se debe pedir al usuario un numero entero, siendo este el tamano del arreglo
// reservar un bloque de memoria con new int[n]
// Usar un bucle para que el usuario ingrese valores enteros
// Mostrar los n valores ingresados
// Liberar la memoria con delete[]
```

```
// Manejar brevemente el caso de tamano invalido <0 mostrando un mensaje y
// saliendo.

#include <iostream>
using namespace std;

int main() {
      int TAM;

          cout << "Bienvenido, por favor, ingrese el tamaño de su arreglo\n";
          cin >> TAM;

        int* datoUser = new int[TAM];

        for (int i = 0; i < TAM; i++) {
            cout << "Ingrese valores " << i + 1 << ": ";
            cin >> datoUser[i];

  }

        for (int i = 0; i < 10; i++) {
            cout << datoUser[i] << " ";

  }

        cout << endl;

        delete[] datoUser;

        return 0;
}
```