```cpp
#include <iostream>
#include <cstdlib>

using namespace std;


/*Define the struct */
struct Arrays {
 char * charPtrArr[20];
 int intArr[20];
};

/*Declare functions */
void deallocAll(struct Arrays&);
void deleteChars(struct Arrays&, int);
int getIndex();
int getMainChoice();
void initArrays(struct Arrays&);
void listDealloc(struct Arrays&);
void printChars(struct Arrays&, int);
void printMainMenu();
void printSubMenu();
char getRandomUpperCaseChar();
void reInitCharPtr(struct Arrays&, int);

main(){
        struct Arrays arrays;

        initArrays(arrays);

        int mainChoice= 0;
        int index = 0;
        int subChoice = 0;
        while(true){
                //Main menu
                mainChoice = getMainChoice();

                switch(mainChoice){
                        case 1:
                                index = getIndex();
                                //Access a pointer
                                if(arrays.charPtrArr[index] == NULL){
                                        reInitCharPtr(arrays, index);
                                }

                                //submenu
                                printSubMenu();
                                cin >> subChoice;

                                //subSwitch
                                switch(subChoice){
                                        case 1:
                                                printChars(arrays, index);
                                                break;
                                        case 2:
                                                deleteChars(arrays, index);
                                                break;
                                        case 3:
                                                break;
                                        default:
                                                break;
                                }
                                break;

                        case 2:
                                //list deallocated memory
                                listDealloc(arrays);
                                break;

                        case 3:
                                //dealloc all
                                deallocAll(arrays);
                                break;

                        case 4:
                                deallocAll(arrays);
                                exit(0);
                                break;

                        default:
                                break;
                }
        }

        return 0;
}

/*deletes all chars of a pointer to char */
void deleteChars(struct Arrays& a, int index){
        delete [] a.charPtrArr[index];
        a.charPtrArr[index] = NULL;
}

/*gets index to access from user*/
int getIndex(){
        int index = -1;
        do{
                cout << "Enter the index to access (0-19)" << endl;
                cin >> index;

                if(index < 0 || index > 19){
                        cout<< "Invalid input. Try again" << endl;
                }
        }while(index < 0 || index > 19);

        return index;
}

/*deletes all pointers*/
void deallocAll(struct Arrays& a){
        for(int i = 0; i < 20; i++){
                deleteChars(a, i);
        }
}

/*gets main menu choice from user*/
int getMainChoice(){
        int choice = -1;
        do{
                printMainMenu();
                cin >> choice;

                if(choice < 1 || choice > 4){
                        cout<< "Invalid input. Try again" << endl;
                }
        }while(choice < 1 || choice > 4);

        return choice;
}

/* returns a random upper case char */
char getRandomUpperCaseChar(){
        return char(rand() % 26 + 65);
}

/*gets sub menu choice from user*/
int getSubChoice(){
        int choice = -1;
        do{
                printSubMenu();
                cin >> choice;

                if(choice < 1|| choice > 4){
                        cout<< "Invalid input. Try again" << endl;
                }
        }while(choice < 1 || choice > 4);

        return choice;
}

/*initializes both arrays*/
void initArrays(struct Arrays& a){

        cout << "Initializing char array..." << endl;

        a.intArr[0] = 2900;

        for(int i = 1; i < 20; i++){
                a.intArr[i] = 2 * a.intArr[i - 1];
        }

        for(int i = 0; i < 20; i++){
                //allocate according to size of int array
                a.charPtrArr[i] = new char[a.intArr[i]];

                for(int j = 0; j < a.intArr[i]; j++){
                        a.charPtrArr[i][j] = getRandomUpperCaseChar();

                }
        }
}

/*lists deallocated indices*/
void listDealloc(struct Arrays& a){
        cout << "List of deallocated char pointers (indices): " << endl;
        for(int i = 0; i < 20; i++ ){
                if(a.charPtrArr[i] == NULL){
                        cout << i << ", ";
                }
        }
        cout << endl;
}

/*prints first ten chars*/
void printChars(struct Arrays& a ,int index){
        for(int i = 0; i < 10; i ++){
                cout << a.charPtrArr[index][i];
        }
        cout << endl;
}

/*prints main menu*/
void printMainMenu(){
        cout << "1. Access a pointer\n"
                << "2. List deallocated memory (index)\n"
                << "3. Deallocate all memory\n"
                << "4. Exit program" << endl;
}

/*prints sub menu*/
void printSubMenu(){
        cout << "1. Print first 10 chars\n"
                << "2. Delete  all chars at this pointer\n"
                << "3. Return to Main menu" << endl;
}

/*renitializes a pointer*/
void reInitCharPtr(struct Arrays& a, int index){
        cout << "This char pointer was previously deleted. Reinitializing.." << endl;
        a.charPtrArr[index] = new char[a.intArr[index]];
        for(int i = 0; i < 10; i++){
                a.charPtrArr[index][i] = getRandomUpperCaseChar();
        }
}
```