

Parcial Criminal

(este parcial es tan difícil que ni siquiera está aprobado por el profe, es completamente ilegal, es un dolor de cabeza y está todo hecho con la teoría del campus, disfruten)

Aclaraciones importantes

- Cuando te pide encontrar el error, son errores de código, por ejemplo, si puse “encontar” en vez de “encontrar” no cuenta como error.
- Escriban sus respuestas aparte y luego comparen con las respuestas, así practican que responder y cómo hacerlo.

1. Dado el siguiente código:

```
10      int a[42]={1,2,3,4,5,6,7,8,9};
11      int b=9;
12      Pila c;
13      inicpila(&c);
14
15      funcion_1(a,b,c);
16      mostrar(&c);
17
50      void funcion_1(int a[], int b, Pila c)
51      {
52          int d=0;
53
54          while(d<b)
55          {
56              apilar(&c,a[d]);
57              d++;
58          }
59      }
60
```

- ¿Qué hace la función 1?
- ¿Qué muestra la pila en la línea 16?

2. Seleccione **la** correcta. Una Función es:

- Una estructura repetitiva.

- b. Un sub-algoritmo encargado de clasificar el código en partes pequeñas con el objetivo de controlar mejor que ve el usuario por pantalla.
- c. Una forma de organizar nuestro código donde las funcionalidades de cada variable se guardan en clases distintas, haciendo el código más legible.
- d. Un sub-algoritmo compuesto de un conjunto de instrucciones que permite resolver una tarea específica y que forma parte del algoritmo principal.
- e. Un sistema de ventanas, iconos, botones, imágenes que permite organizar el código.

3. Cual de estas se considera una buena práctica en la programación en c:

- a. Modularización.
- b. Polimorfismo.
- c. Nombres significativos de identificadores y funciones.
- d. Encapsulamiento.
- e. Funciones que se encarguen de hacer 2 o más tareas.

4. Dado el siguiente código con los nombres y los prints modificados:

```
int crg_str(char a[], int d)
{
    char b='s';
    int c=0;

    while(b=='s' && c<d)
    {
        printf("\nMisterio1: ");
        fflush(stdin);
        scanf("%c", &a[c]);

        c++;

        printf("\nMisterio2: ");
        fflush(stdin);
        scanf("%c", &b);
    }

    return c;
}
```

- a) ¿Qué hace el código?.
- b) ¿Funciona correctamente?.

5. ¿Qué es lo que hace el siguiente código?(funciona correctamente).

```
77  int funcion_punto_5(int a[], int b)
78  {
79      int c=0;
80      int d=b-1;
81      int e=0;
82
83      while(c<d && e==0)
84      {
85          if(a[c]!=a[d])
86          {
87              e=1;
88          }
89          c++;
90          d--;
91      }
92      return e;
93  }
94
```

6. Explique, con sus palabras, el método de ordenación por inserción.

7. Encuentre los errores en este código:

```
98  int carga_arr(int* a[], int dim)
99
100  char yes='s';
101  int i=0;
102
103  while(yes=='s' && i<dim)
104  {
105      printf("\nIngrese un numero al arreglo: ");
106      fflush(stdin);
107      scanf("%i", (*a[i]));
108
109      i++;
110
111      printf("\nDesea continuar? s/n ");
112      fflush(stdin);
113      scanf("%c", &yes);
114  }
115  return i;
116
117
```

8.a. ¿Qué hace esta función?

b. ¿Funciona correctamente este código?

```
117  int busca_string(char m[a][c], char aux[])
118  {
119      int i=0;
120      int f=-1;
121
122      for(i=0; i<a; i++)
123          if(strcmp(m[i], aux)==0)
124              f=i;
125
126      return f;
127  }
```

9. ¿Qué son los válidos? ¿Y la dimensión?

10. Preguntas de arreglos. Seleccione verdadero o falso, justifique o corrija las falsas.

- a. El de ints debe cargarse por completo mientras que el de strings no.
- b. El de strings tiene dos dimensiones y el de int una.
- c. El arreglo de ints no puede ordenarse por selección ni inserción.
- d. El arreglo de strings no puede contener números.
- e. El arreglo de strings tiene funciones propias para compararse o copiarse y el de ints también.
- f. El arreglo de strings puede ser ordenado por inserción y el int por selección.

11. Dado el siguiente código:

```
int nuevo=9;
int aux;
int viejo=8;
viejo=funcion_punto11(nuevo,&aux, viejo);
```

```
int funcion_punto11(int nuevo, int* aux, int viejo)
{
```

```
(*aux)=nuevo;  
nuevo=viejo;  
viejo=aux;  
return viejo;  
}
```

¿Que valor vería si hago un print de las variables nuevo, viejo y aux despues de pasar por la función “funcion_punto11”?

- a. nuevo=9,viejo=(*dicc. memoria*), aux=9;
- b. nuevo=8,viejo=9, aux=8;
- c. nuevo=8,viejo=9, aux=(*dicc. memoria*).
- d. nuevo=9,viejo=(*dicc. memoria*), aux=9;

12. Verdadero o falso funciones. Justifique o corrija las falsas:

- a. Las funciones son una forma de ahorrar tiempo.
- b. La única forma de guardar el dato que obtuve de la función es retornandolo.
- c. Hacer todo el código en el main,sin usar nunca una función es imposible.
- d. Una función si o si debe tener parámetros que son traídos desde el main.
- e. No pueden crearse variables de tipo puntero dentro de las funciones.
- f. Es una buena práctica que las funciones realicen una sola acción. Ej. una función de cargar solo debe cargar.
- g. Al trabajar con funciones de arreglos previamente cargados, una función nunca debe retornar un dato que no se encuentre en el arreglo.

13. Responda v o f:

Fuera del main únicamente pueden declararse funciones o constantes globales.

14. Explique, con sus palabras, el método de ordenación por selección.

15. Responda v o f:

Una matriz solo se puede recorrer con la estructura repetitiva for.

16. Explique cada una de las funciones de Pila.(son 7)

17. Dada la siguiente función, explique con sus palabras cuál es su funcionamiento

```
158
159  int funcion_punto_17(char s[filas][col], int v, char x[filas])
160  {
161      int i=0;
162      int j=0;
163
164      while(i<v)
165      {
166          if(strcmpi(s[i],x)==0)
167          {
168              while(j<v)
169              {
170                  strcpy(s[j],s[j+1]);
171                  j++;
172              }
173              v--;
174          }
175          i++;
176          j=i;
177      }
178      return v;
179  }
180
```

18. Que retorna una función void:

- a. Retorna un tipo de dato **default**.
- b. Retorna todos los parámetros.
- c. No retorna nada.
- d. Retorna un tipo de dato **void**.

19. ¿Qué es un puntero?

20. Responda verdadero o falso:

¿Puede haber un puntero de un puntero?. Ej. int** a;

Respuestas

1.a. Apila todos los elementos válidos de un arreglo en una pila pasada por copia.

b. No muestra nada porque al estar pasada por copia, la pila original del main queda vacía.

2.d.

3. a - c.

4. a. Carga un arreglo de caracteres (no uno de strings).

b. A pesar de que el nombre podía resultar confuso (esa era la trampa), carga el arreglo correctamente.

5. la función te devuelve 1 si el arreglo NO es capicúa y 0 si el arreglo lo es.

6. El método de ordenamiento por inserción consiste en tomar un elemento de la parte desordenada del arreglo e ir insertando en la parte ordenada de este. para no sobrescribir siempre el mismo dato, lo que se hace es, en cada iteración del bucle que utilicemos pasarle la posición donde queremos insertarlo (Esto puede variar, no se asusten si no ponen lo que yo puse acá).

7. 1. La función no abre ni cierra llaves.

2. en los parámetros de la función pide un "int* a[]", esto es un error porque los arreglos ya son punteros por definición.

3. En el "scanf" de la línea 113 está el & donde debería estar el % y viceversa.

8. a. Buscas un string en un arreglo de strings y devuelve su posición.

b. Sí. (Si tu estructura repetitiva o si tu if solo es una línea de código no hace falta abrir y cerrar llaves, pruebenlo si quieren)

```
int busca_string(char m[a][c], char aux[])
```

```
{
    int i=0;
    int f=-1;

    for(i=0; i<a; i++)
        if(strcmp(m[i],aux)==0)
            f=i;

    return f;
}
```

9. Los validos son la cantidad de elementos que me importan en el arreglo y dimensión es el tope máximo de elementos que puede albergar un arreglo, se define al momento de crearlo.

10.a.F. No hace falta que ninguno se cargue por completo.

- b.V.
- c. F. El arreglo de ints puede ordenarse por selección e inserción.
- d. F. El arreglo de strings puede contener cualquier tipo de carácter, incluido números, sin embargo no es una forma práctica de guardarlos.
- e. F. El arreglo de strings tiene funciones propias para compararse o copiarse y el de ints no.
- f. V. (como aclaración, la pregunta no dice que no se pueda usar el otro ordenamiento).

11. a.

12.

- a. V.
- b. F. para guardar un dato obtenido de una función puede retornar o, usar un puntero desde el principio, de esta manera uso la dirección de memoria de la variable y no necesito retornarlo.
- c. F. Hacer todo el código en el main, sin usar nunca una función es posible, lo hace incomodo, tu código pelagra de tener errores, hace que repitas código muchas veces, pero es posible.
- d. F. Una función puede o no tener parámetros que son traídos desde el main.
- e. F. pueden crearse variables de tipo puntero dentro de las funciones.
- f. V.
- g. F. No hace falta que, al trabajar con funciones de arreglos previamente cargados, esta no deba retornar un dato que no se encuentre en el arreglo porque por ejemplo, al buscar algo dentro de un arreglo puedo retornar -1 si no lo encuentro.

13. Falso.

14. El ordenamiento por selección tiene 2 funciones la primera busca un dato específico en el arreglo, generalmente el menor, y retorna su posición, mientras que la otra función hace el cambio entre el menor y la primera posición del arreglo, luego se aumenta el indicador de posición para no pisar el dato menor, que está en la primera posición.

15. Falso.

16. apilar -> apila un elemento al tope de la pila.
 desapilar -> quita un elemento del tope de la pila.
 mostrar -> muestra toda la pila.
 leer -> pide al usuario que ingrese un elemento a la pila.
 inicpila-> inicializa la pila.
 pilavacia -> devuelve 1 si la pila está vacía.
 tope -> devuelve el tope de la pila.

17. La función busca un elemento en el arreglo, si lo encuentra, borra el elemento, no deja el espacio y resta uno en sus válidos.

18. c.

19. Una variable que guarda la dirección de memoria de un tipo de dato específico.

20. Verdadero(se llama puntero doble).