

DATOS DA PROBA

TIPO DE PROBA	Actividade <input type="checkbox"/> Exame <input checked="" type="checkbox"/>
CURSO	IFCT0609 – PROGRAMACIÓN DE SISTEMAS INFORMÁTICOS (2019/001293)
M. FORMATIVO	MF0965_3: Desenvolvemento de Software baseado en tecnoloxías orientadas a compoñentes
U. FORMATIVA	UF1289: Desenvolvemento de elementos software con tecnoloxías baseadas en compoñentes
CRITERIOS AVALIABLES	E1: Avaliación teórico-práctica Proba teórica e de respostas curtas
INSTRUCCIÓNS	

DATOS ALUMNO		CALIFICACIÓN
DNI		E1
SINATURA		

INSTRUCCIÓNS

O alumno dispón de 2 horas para responder ás seguintes preguntas tipo test. Cada pregunta contestada de maneira correcta será valorada con 0.5 puntos. 3 preguntas mal contestadas restarán unha ben contestada. As preguntas en branco non suman nin restan. Para superar esta proba o alumno debe contestar correctamente polo menos 10 das 20 preguntas plantexadas. Valoración total: 10 puntos sobre 10. Cada pregunta só ten unha resposta correcta

CONTIDO TEÓRICO DA PROBA E1

1. Resposta breve: ¿Qué diferencias fundamentales existen entre un bucle for y un while?

¿Pueden actuar uno en lugar del otro? **RAZONA TU RESPUESTA**

Ambos bucles se pueden intercambiar entre sí, pero mientras que en el for controlas el índice del bucle en la zona de declaración, en el while lo tienes que controlar manualmente.

2. ¿Qué salida tiene el código que se muestra en pantalla?

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5 int main(){
6     int i=0;
7     char nombre[10];
8
9     for(i=0;i<strlen(nombre);i++){
10         printf("Introduce la letra %d del nombre: ",i+1);
11         scanf("%c",nombre[i]);
12     }
13     printf("El nombre introducido es: %s",nombre);
14
15     return 0;
16 }
```

- Ninguna. Da un error de compilación porque se utiliza strlen(nombre).
- Pide uno a uno 10 caracteres (se almacenan en la variable nombre) para acabar mostrando la variable nombre.
- El bucle for no hace nada, se muestra la variable nombre con su valor inicial.**
- Da error en la ejecución. Bucle infinito.

3. Explica brevemente la diferencia entre la función strlen y la función sizeof. Puedes basarte en el siguiente ejemplo. Indica que devuelve el código en una posible ejecución.

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5 int main(){
6     char cadena[100]="Examen de PSI";
7
8     printf("%ld",strlen(cadena));
9     printf("\n%ld",sizeof(cadena));
10
11     return 0;
12 }
13
```

La función `strlen` devuelve la longitud de una cadena (13 en el ejemplo) y `sizeof` el tamaño reservado (100 en el ejemplo)

4. ¿Qué realiza el siguiente código?

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5 int main(){
6     int i;
7     char cadena[100]="Examen de PSI";
8     for(i=0;cadena[i]!='\0';i++){
9         cadena[i]=5;
10    }
11    printf("%s",cadena);
12    return 0;
13 }
14
```

- a) No realiza ninguna acción ya que la asignación de la línea 9 es errónea.
- b) Recorre la cadena hasta el carácter final y sustituye cada uno por un 5.
- c) Recorre la cadena hasta el carácter final y sustituye cada uno por la frase "Examen de PSI".
- d) Recorre la cadena hasta el carácter final y sustituye cada uno por el símbolo del trébol.

5. Hemos realizado dos modificaciones en el código anterior quedando de la siguiente manera. ¿Qué realiza el código ahora?

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5 int main(){
6     int i;
7     char cadena[100]="Examen de PSI";
8     for(i=0;cadena[i]!='\0';i++){
9         printf("Original: %c-->",cadena[i]);
10        cadena[i]='5';
11        printf("Nuevo: %c\n",cadena[i]);
12    }
13    printf("\n%s",cadena);
14    return 0;
15 }
```

- a) Recorrer una cadena hasta el final e ir mostrando los cambios que se realiza en cada vuelta del bucle (cada letra se sustituye por un trébol). Al final muestra la cadena (13 tréboles).
- b) Recorrer una cadena hasta el final e ir mostrando los cambios que se realiza en cada vuelta del bucle (cada letra se sustituye por un 5). Al final muestra la cadena (13 cincos).
- c) El código es erróneo. Para hacer esa asignación deberíamos utilizar `strcpy(cadena[i],"5")`.
- d) Obtenemos un error de compilación por desbordamiento del bucle `for`.

6. ¿Qué función realiza el siguiente código?

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5 int main(){
6     const int TAM=5;
7     int i;
8     int miArray[TAM];
9     for(i=0;i<TAM;i++){
10         printf("Introduce el elemento %d: ",i+1);
11         scanf("%d",&miArray[i]);
12     }
13     return 0;
14 }
15

```

- Obtenemos un error por utilizar la variable TAM y no inicializar el array manualmente.
- Obtenemos desbordamiento del bucle, ya que no se utiliza la función strlen para controlar el final del for.
- Estamos inicializando un array de caracteres al carácter trebol.
- Estamos inicializando un array de enteros pidiendo cada uno de sus números por teclado.

7. Hemos mejorado el código anterior añadiendo una funcionalidad más. ¿Qué realiza ahora el código?

```

1 #include <stdio.h>
2
3 int main(){
4     const int TAM=5;
5     int i;
6     int miArray[TAM];
7     for(i=0;i<TAM;i++){
8         printf("Introduce el elemento %d: ",i+1);
9         scanf("%d",&miArray[i]);
10    }
11    printf("Mi array: ");
12    for(i=0;i<TAM;i++)
13        printf("%d ",miArray[i]);
14
15    return 0;
16 }

```

- Muestra el mismo error de ejecución que el ejercicio anterior respecto a la variable TAM.
- Inicializamos el array pidiendo los números por teclado y luego los mostramos.
- Inicializamos el array pidiendo los char por teclado y luego se muestran.
- Inicializamos 2 arrays diferentes: uno con caracteres pedidos por teclado y otro a partir de la variable i.

8. Vamos a realizar un programa que me permita comparar si dos cadenas son iguales. ¿La siguiente base sería correcta? **RAZONA TU RESPUESTA**

```

1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     char cadena1[10]="Pepe", cadena2[10]="Pepe";
6     int i=0, aciertos=0;
7     for(i=0;i<=strlen(cadena1);i++){
8         if(cadena1[i]==cadena2[i])
9             aciertos++;
10    }
11    if (aciertos==strlen(cadena1))
12        printf("Las cadenas son iguales.");
13    else printf("Las cadenas son diferentes");
14    return 0;
15 }
16

```

No. Para comparar 2 cadenas (línea 8) hay que utilizar strcmp (librería string.h)

9. Respecto al ejercicio anterior: si has contestado que el código es correcto, modifica el for para que también cuente los fallos. Si has contestado que el código es erróneo, aporta como debería ser la solución. **SOLAMENTE TIENES QUE APORTAR LAS LÍNEAS DEL BUCLE FOR (DESDE LA 7 HASTA QUE TERMINE CON LA MODIFICACIÓN).**

```
for(i=0;i<strlen(cadena);i++){
    if(strcmp(cadena1,cadena2)==0)
        aciertos++;
}
```

10. ¿Qué función realiza el código aportado en la siguiente imagen?

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     const int TAM=20;
6     char cadena[TAM], aux[TAM];
7     int i;
8     printf("Introduce una cadena: ");
9     gets(cadena);
10    for(i=0;i<strlen(cadena);i++){
11        aux[i]=cadena[i];
12    }
13    aux[i]='\0';
14    return 0;
15 }
```

- a) Realiza una copia carácter a carácter de la cadena introducida por teclado en aux.
- b) Realiza una comparación de una cadena con los valores de i hasta, como indica la línea 13, el '\0'
- c) Da un error en la ejecución ya que el bucle está mal controlado.
- d) Da un error en compilación ya que los dos array de tipo cadena no están inicializados.

11. ¿Qué función realiza el código aportado en la siguiente imagen?

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     int i=0, num, posi=0, encontrado=0, miArray[6]={1,5,7,9,12,6};
6     printf("Introduce un numero: ");
7     scanf("%d",&num);
8     for(i=0;i<(sizeof(miArray)/sizeof(int));i++){
9         if(miArray[i]==num){
10             encontrado=1;
11             posi=i;
12         }
13     }
14     if(encontrado==1) printf("\nPosicion: %d",posi);
15     else printf("\nNo esta");
16     return 0;
17 }
```

- a) Realiza la sustitución de un carácter del array "miArray" por otro introducido por teclado. Indica la posición del cambio.
- b) Realiza la búsqueda de un elemento de "miArray" y si lo encuentra, muestra su posición.
- c) Ordena un array posición a posición
- d) Muestra un error ya que miArray[i]==num no funciona; debemos usar strcmp.

12. Necesitamos recorrer un array de caracteres hasta el final para contar cuantas 'a' o 'A' tenemos. ¿Sería posible con la siguiente opción? **RAZONA BREVEMENTE TU RESPUESTA**

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     int i=0,aes=0;
6     char miArray[100]="Este es mi array";
7     while(miArray[i]!='\0'){
8         if(miArray[i]=='a' || miArray[i]=='A')
9             aes++;
10    }
11    return 0;
12 }
```

- a) Sí ->
- b) No -> La i no avanza al acabar el while. Se produce un bucle infinito

13. ¿Qué función realiza la siguiente porción de código?

```

1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     int i=0, j=0;
6     char miArray[20]="Examen de PSI", aux[20];
7     while(miArray[i]!='\0'){
8         if(miArray[i]!=' '){
9             aux[j]=miArray[i];
10            j++;
11        }
12        i++;
13    }
14    printf("%s",aux);
15    return 0;
16 }

```

- a) Vuelca la cadena en aux eliminando el carácter '\0' y después la muestra.
- b) Vuelca la cadena en aux eliminando el carácter espacio y después la muestra.**
- c) Produce un error de compilación ya que aux[j]=miArray[i] debería hacerse con strcmp.
- d) Vuelca en la cadena aux una cadena solo con espacios y después la muestra.

14. Si al while le quito las llaves de inicio y final, ¿obtendría el mismo resultado? Razona tu respuesta.

- a) Sí ->
- b) No-> El while incluye 2 instrucciones (el if y el i++).**

15. ¿Qué función tiene la siguiente porción de código?

```

1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     int a=1, b=0;
6     while(a!=0 && b<5){
7         printf("Vamos!: ");
8         scanf("%d",&a);
9         b++;
10    }
11    return 0;
12 }

```

- a) Pedir letras mientras introduzcamos un número entre cero y 5.
- b) Pedir números mientras sean menores que cinco o el cero.
- c) Pedir números mientras no se introduzca el cero o no se hagan 5 intentos.**
- d) El programa no funciona correctamente ya que deberíamos usar funciones de cadena.

16. ¿Qué salida tiene la siguiente porción de código?

```

1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     char asignatura[10]="PSI",opc[10]="";
6     while(strcmp(asignatura,opc)!=0){
7         printf("Vamos!: ");
8         gets(opc);
9     }
10    return 0;
11 }

```

- a) Muestra un error; deberíamos utilizar la función strcpy ya que se trata de strings.
- b) En la segunda vuelta muestra un error porque captura un '\n' del buffer y lo reproduce.
- c) No se pueden comparar los dos array porque las inicializaciones son erróneas.
- d) Pide nombres hasta que introduzcas PSI.**

17. El siguiente fragmento de código no funciona correctamente. ¿Qué está pasando?

NOTA: no hay problemas de llaves, ni de librerías ni de punto y coma.

```

1 #include <stdio.h>
2 #include <string.h>
3 int main(){
4     int num=1, a=0, b=0;
5     while(num!=0){
6         printf("Introduce un numero: ");
7         scanf("%d",&num);
8         if(num%2==0)
9             a++;
10        else b++;
11    }
12    return 0;
13 }

```

Intenta pedir números hasta introducir un cero y guarda en la variable a los pares y en la variable b los impares. No funciona correctamente porque num está inicializada a cero

18. ¿Qué modificación añadirías al ejercicio anterior (respetando su estructura básica) para que funcione correctamente? Una vez que está correcto, ¿Qué tarea realiza el código?

Inicializar num a 1 para entrar en el bucle

19. El siguiente código está incompleto. Suponemos que el valor de la cadena puede ser cualquier palabra, es decir, no tiene un largo fijo. Además, queremos sustituir el símbolo € por la letra E. ¿Qué falta en los huecos marcados como "XXXXXX" para que el programa funcione correctamente?

```

1 #include <stdio.h>
2 #include <string.h>
3 int main(){
4     int i;
5     char cadena[50]="€XAM€N";
6     for(i=0;XXXXXX='\0';i++){
7         if(XXXXXX)
8             cadena[i]='E';
9     }
10    return 0;
11 }

```

- a) Sizeof(cadena), strcmp(cadena, '€')
- b) Sizeof(cadena), strcmp('€', cadena)
- c) Strlen(cadena), strcmp(cadena, '€')
- d) Cadena[i]!='\0', cadena[i]=='€'

20. ¿Qué función realiza el siguiente código? **RAZON TU RESPUESTA**

```

1 #include <stdio.h>
2 #include <string.h>
3 int main(){
4     char a='s', b='n';
5     while(tolower(a)!='n' || tolower(b)!='s'){
6         printf("Sigo?(s/n): ");
7         scanf("%c",&a);
8         fflush(stdin);
9         if (tolower(a)=='n'){
10            printf("Seguro?(s/n)");
11            scanf("%c",&b);
12            fflush(stdin);
13            if(tolower(b)=='s')
14                printf("OK\n");
15        }
16    }
17    return 0;

```

El programa pide continuar hasta decir que no quiero seguir y en la posterior pregunta, seguro, digo también que sí. En cuanto se cumplen estas dos condiciones, el programa termina.

21. **BONIFICACIÓN.** Esta pregunta computará a cambio de una mal contestada. Dado el siguiente código, ¿Cómo puedo sumar 5 a la variable numero? **RAZONA TU RESPUESTA**

```
1 #include <stdio.h>
2 #include <string.h>
3 int main(){
4     char numero[5]="1234";
5
6
7     return 0;
8 }
```

Transformamos la variable string en numero con atoi y le sumamos 5.

```
1 #include<stdio.h>
2 #include<string.h>
3
4 int main(){
5
6     char numero[5]="1234";
7     int resultado=0;
8     resultado=atoi(numero);
9     printf("Sumando cinco: %d",resultado+5);
10     return 0;
11 }
```