

Question 1 of 10

Haga coincidir las siguientes opciones:

1. []	Describen acciones algorítmicas que pueden ser ejecutadas	A. dato
2. []	Conjunto de datos unidos por operadores que tienen un único resultado	B. expresión
3. []	Los parámetros cambian sus contenidos	C. if, if else
4. []	Información con la que un programa trabaja	D. sentencia
5. []	Tipos de operadores	E. paso por referencia
6. []	Tipos básicos de estructuras de control	F. asignación, relación y lógico

Question 2 of 10

Indicar cuantas veces se despliega Hola en pantalla.

```
int i;  
for ( i=1 ; i<5 ; i+=1 ) {    printf( "Hola\n" );    }
```

- ☐ A) Hola, 4 veces
- ☐ B) Hola, 5 veces
- ☐ C) Hola, 0 veces
- ☐ D) Hola, 3 veces

Question 3 of 10

En la operación $!(A \ \&\& \ B)$, si A es True (Verdadero) y B es False (Falso), la respuesta es:

- ☐ A) No se puede saber
- ☐ B) Verdadera
- ☐ C) No es una operación válida
- ☐ D) Falsa

Question 4 of 10

Según el código siguiente, ¿cuál es la forma correcta de ingresar 3 datos (un dato flotante, un caracter para la operación, y el otro dato flotante) para que se realice la suma de ambos flotantes?

```
int main() {  
    float a,b,c;    char Op;  
  
    printf("Digite la operación a realizar :");  
    scanf("%f%c%f", &a, &Op, &b);  
  
    switch(Op) {  
        case '+':  
  
            c=Suma(a,b);  
            printf("La suma de %f + %f = %f\n",a,b,c);  
            break;  
        .....  
    }
```

- ☐ A) 2.7 <enter> + <enter> 5.1 <enter>
- ☐ B) 2.7 + 5.1 <enter>
- ☐ C) 2.7 <espacio> + <espacio>5.1<enter>

Question 5 of 10

Para el producto de dos números complejos, siendo el primero $C1 = a + bi$, y el segundo $C2 = c + di$, ¿cómo se calcula la parte Resultante real **Rr**, y cómo se calcula la parte Resultante imaginaria **Ri**?

```
void ProductoComplejo(float a, float b, float c, float d){  
    float Rr, Ri;  
    Rr = _____  
  
    Ri = _____  
  
    if(Ri<0)  
        printf("R=%f %fi\n",Rr,Ri);  
    else  
        printf("R=%f + %fi\n",Rr,Ri);  
}
```

☐ **A)** $Rr=(a*d)+(b*c);$

$Ri=(b*d)+(b*c);$

☐ **B)** $Rr=(a*c)*(b+d);$

$Ri=(a-d)-(a*b);$

☐ **C)** $Rr=(a*d)-(b*c);$

$Ri=(a*d)+(b/d);$

☐ **D)** $Rr=(a*c)-(b*d);$

$Ri=(a*d)+(b*c);$

Question 6 of 10

Considerando la siguiente función que calcula las raíces de una ecuación de segundo grado con la fórmula general, ¿qué operaciones faltan en los espacios en blanco?

```
void funcion(void){
    int a, b, c, dis;
    float x1, x2;
    printf("Ingrese a ");scanf("%d",&a);
    printf("Ingrese b ");scanf("%d",&b);
    printf("Ingrese c ");scanf("%d",&c);

    dis = _____ i _____;
    if(_____ ii _____)
        printf("Indeterminacion de formula.\n");
    else if(_____ iii _____)
        printf("Raices negativas.\n");
    else{
        x1 = _____ iv _____;
        x2 = _____ v _____;
        printf("x1=%f\n",x1);
        printf("x2=%f\n",x2);
    }
}
```

- ☐ A) i -> $\text{pow}(b,2)-4*a*c$
ii -> $a \neq 0$
iii -> $\text{dis} > 0$
iv -> $(-b+\text{sqrt}(\text{dis}))/ (2*a)$
v -> $(-b-\text{sqrt}(\text{dis}))/ (2*a)$
- ☐ B) i -> $\text{pow}(b,2)-4*a*c$
ii -> $a < 0$
iii -> $\text{dis} == 0$
iv -> $(-b+\text{sqrt}(\text{dis}))/ (2*a)$
v -> $(-b-\text{sqrt}(\text{dis}))/ (2*a)$
- ☐ C) i -> $\text{sqrt}(\text{pow}(b,2)-4*a*c)$
ii -> $a > 0$
iii -> $\text{dis} \neq 0$
iv -> $(-b+\text{sqrt}(\text{dis}))/ (2*a)$
v -> $(-b-\text{sqrt}(\text{dis}))/ (2*a)$
- ☐ D) i -> $\text{pow}(b,2)-4*a*c$
ii -> $a == 0$
iii -> $\text{dis} < 0$
iv -> $(-b+\text{sqrt}(\text{dis}))/ (2*a)$
v -> $(-b-\text{sqrt}(\text{dis}))/ (2*a)$

Question 7 of 10

Seleccione la opción que rellene correctamente los espacios del código siguiente, para que se muestre la tabla de la función Ecua1() desde un inicio, hasta un final y con incrementos dados por el usuario.

```
void LeeDts(_____, _____, _____);
float Ecua1(float x );
int main ()
{ int i;
  float X,Y[100];
  float xi, xf, inc;
  LeeDts(_____, _____, _____);
  X = xi;  i=0;
  printf("\n X \t\t Y \n");
  while (X <= xf) {
    if( X != 0 ){
      Y[i] = Ecua1(X);
      printf("%f\t%f\n",X,Y[i]);
    }
    else
      printf("Se indetermina con x = %f\n",X);
    _____;
  }
  return 0;
}

void LeeDts(_____, _____, _____){
  printf("Ingresa xi ");scanf("%f",xi);
  printf("Ingresa xf ");scanf("%f",xf);
  printf("Ingresa inc ");scanf("%f",inc);
}
float Ecua1(float x){
  return (7.5*x*x + (3/x) + 0.4 );
}
```

-
- ☐ A) float *xi, float *xf, float *inc
 &xi, &xf, &inc
 X=X+i; inc++
 float *xi, float *xf, float *inc
- ☐ B) float *xi, float *xf, float *inc
 xi, xf, inc
 X=X+inc; i++
 float *xi, float *xf, float *inc
- ☐ C) float *xi, float *xf, float *inc
 &xi, &xf, &inc
 X=X+inc; i++
 float *xi, float *xf, float *inc
- ☐ D) float xi, float xf, float inc
 &xi, &xf, &inc
 X=X+inc; i++

float xi, float xf, float inc

Question 8 of 10

Seleccione la opción que rellene correctamente los espacios en blanco de la siguiente función, que despliega los meses con sus días, para todo el año.

```
void cal( _____ );
int main(){
char meses [12][11]= {"Enero", "Febrero", "Marzo",
                      "Abril", "Mayo", "Junio",
                      "Julio", "Agosto", "Septiembre",
                      "Octubre", "Noviembre", "Diciembre"};
int dias [12]={31,28,31,30,31,30,31,31,30,31,30,31};

cal(_____);
system("pause");
return 0;
}

void cal( _____ )
{ int i;
  for (i=0;i<12;i++)
    printf( _____ );
}
```

- ☐ **A)** char meses [12][11], int dias [12]
meses;dias
char meses [12][11]; int dias [12]
" %s tiene %d dias. \n", meses[i], dias[j]);
- ☐ **B)** int dias [12], char meses [12][11]
días.meses
int dias [12], char meses [12][11]
" %c tiene %d dias. \n" meses[i], dias[i]);
- ☐ **C)** int dias [12], char meses [12][11]
dias,meses
int dias [12], char meses [12][11]
" %s tiene %d dias. \n", meses[i], dias[i]);
- ☐ **D)** char meses [12][11], int dias [12]
meses,días
char meses [12][11]. int dias [12]
" %c tiene %d dias. \n" meces[i], dias[i]);

Question 9 of 10

Seleccione la opción que rellene correctamente los espacios en blanco de la siguiente función, que calcula los espacios que tiene una oración o cadena de caracteres.

```
int Funcioin1( _____ )
{ int cuantos=0;
  int i=0;
  while( _____ ) {

    if( _____ )
      _____
    i++;
  }
  return _____
}
```

-
- ☐ A) char Cadena[50]
Cadena[i] != '\0'
Cadena[i] == ' '
{cuantos++;}
cuantos;
- ☐ B) char Cadena[]
Cadena[i] != '\0'
Cadena[i] == ' '
cuantos--;
cuantos;
- ☐ C) char Cadena[]
Cadena[i] != '\0'
Cadena[i] <= ' '
cuantos++;
i ;
- ☐ D) char Cadena[50]
Cadena[i] != '\0'
Cadena[i] <= ' '
{cuantos--;}
i ;

Question 10 of 10

Seleccione la opción que rellene correctamente los espacios en blanco de la siguiente función, que despliegue la producción semanal de medicamentos, en forma de matriz; los días de la semana son columnas, las filas son los medicamentos. Considere:

```
char DIAS[5][11] = {"Lunes", "Martes", "Miércoles", "Jueves", "Viernes"};
char *Medicina[5] = {"Ampicilina", "Penicilina",
                    "Acetaminofen", "Paracetamol", "Vontrol"};
float Datos[5][5] = { {100,200,300,400,500 },
                     {110,270,340,420,590 },
                     {120,280,350,430,580 },
                     {130,290,360,440,570 },
                     {140,230,370,450,560 } };

void VerMatriz( _____ ) { int i,j;
    printf("PRODUCCION DE MEDICAMENTOS POR DIA DE LA SEMANA\n\n ");
    printf("\n\t ");
    for(i=0; i<5; i++)
        _____ ;
    printf("\n");
    for(i=0; i<5; i++)
    {
        _____ ;
        for(j=0; j<5; j++)
            _____ ;
    }
    printf( "\n");printf( "\n");
}
```

- ☐ A) char D[5][11], char *Med[5], float dats[5][5]
printf("%12s" , D[i])
printf("%-13s", Med[i])
printf("%11.1f", dats[i][j])
- ☐ B) char D[5][11], float Dat[5][5], char *Med[5]
printf("%12s" , D[i])
printf("%11.1f", dat[i][j])
printf("%-13s", Med[i])
- ☐ C) char *Med[5], float Dat[5][5], char D[5][11]
printf("%-13s", Med[i])
printf("%12s" , D[i])
printf("%11.1f", Da[i][j])
- ☐ D) char D[11][5], char *Med[11], float dats[11][5]
printf("%12s" , D[i])
printf("%-13s", Med[i])
printf("%11.1f", dats[i][j])