

1. Indica la salida de este programa

```
main () {  
int x = 5, y = 10;  
    int *xPtr = NULL;  
    int *yPtr = NULL;  
    xPtr = &x;  
    yPtr = &x;  
    *xPtr = *xPtr + 1;  
    (*yPtr)--;  
printf ("x: %d\n",x);  
printf ("y: %d\n",y);  
system("pause");  
return 0;}
```

2. Indica la salida de este programa

```
main() {  
int n;  
int * ptr;  
n=8;  
    printf("ptr es %p \n", ptr);  
    ptr=&n;  
    printf("n es %d\n", n);  
    printf("&n es %p\n", &n);  
    printf("*ptr es %d\n", *ptr);  
    printf("ptr es %p\n", ptr);  
    n--;  
    *ptr=5;  
    printf("n es despues de n-- %d\n", n);  
    printf("&n es %p\n", &n);  
    ptr++;  
printf("ptr es despues de ptr++ %p\n", ptr);  
printf("*ptr es %d\n", *ptr);  
system("Pause");  
return 0;}
```

3. ¿A qué dirección apuntan m y n al finalizar el programa?

```
main(){  
int a, b, *p, *q, *r;  
char c, d, *m, *n;  
    b=7;  
    a=5;  
    q=&a;  
    p=q;  
    *p=b;  
    printf("%d", *q);  
    r=&b;  
    *r=*q;  
    printf("a=%d, b=%d, *p=%d, *q=%d, *r\n", a, b, *p, *q, *r);  
    c='A';  
    d='B';  
    m=&c;  
    n=&d;  
    *m=*n;  
system("Pause");  
return 0;  
}
```

4. Indica cuál es la salida antes de ejecutarlo.

```
main(){  
int *p;  
int a=1, b=2;  
int **s;  
    p=&a;  
    b=*p+1;  
    s=&p;  
printf("p=%d, *p=%d, &p=%d\n",p,*p,&p);  
printf("s=%d, *s=%d, &s=%d\n",s,*s,&s);  
*s=&b;
```

```

printf("Se ejecutó *s=&b y ahora es\n");
printf("s=%d, *s=%d, &s=%d\n",s,*s,&s);
**s=79;
printf("a=%d, &a=%d\n",a,&a);
printf("b=%d, &b=%d\n",b,&b);
system("Pause");
return 0;}

```

5. ¿qué hace este programa?

```

main()
{ void *generico;
  int *punt_int, x=3;
  char *punt_char = "Puntero a caracteres";
  punt_int=&x;
  printf("El valor apuntado por punt_int es %d (valor de x)\n", *punt_int);
  generico=punt_char;
  printf("\n\ngenerico (tipo void) apunta a objeto char: %s", generico);
  printf("\n\nImprimo ahora generico como char:\n\n");
  puts((char *)generico);
  system("Pause");
  return 0;}

```

6. analiza y escribe que emite en cada printf // Reemplaza, en los casos de direcciones de memoria, el especificador %d por el %p.

```

#include <stdio.h>
#include <stdlib.h>
main() {
  int a,b,*pta,**ptb;

  pta=&a;
  ptb=&pta;
  *pta=8;
  printf("ptb es: %d\n\n",ptb);
  printf("Contenidos de *ptb después de &pta es: %d\n\n",*ptb);
  printf("Contenidos de **ptb después de &pta es: %d\n\n",**ptb);
  **ptb=**ptb+3;
  printf("Contenidos de **ptb es: %d\n\n",**ptb);
  printf("Contenidos de a es: %d\n\n",a);
  *ptb=NULL;
  printf("Contenidos de a después de *ptb=NULL, es: %d\n\n",a);
  printf("Contenidos de *ptb después de NULL es: %d\n\n",*ptb);
  b=8;
  pta=&b;
  printf("Contenidos de *ptb después de pta=&b, es: %d\n\n",*ptb);
  printf("Contenidos de a es: %d\n\n",a);
  printf("Contenidos de b después de pta=&b, es: %d\n\n",b);
  printf("Contenidos de *pta después de pta=&b, es: %d\n\n",*pta);
  printf("Contenidos de **ptb después de pta=&b, es: %d\n\n",**ptb);
  printf("ACA Contenidos de *ptb después de pta=&b, es: %d\n\n",*ptb);
  **ptb=a+10;
  printf("Contenidos de **ptb después de **ptb=a+10, es: %d\n\n",**ptb);
  printf("Contenidos de *pta después de **ptb=a+10, es: %d\n\n",*pta);
  printf("Contenidos de a después de **ptb=a+10, es: %d\n\n",a);
  printf("Contenidos de b después de **ptb=a+10, es: %d\n\n",b);
  printf("La suma de los contenidos de pta + ptb es: %d\n\n",(*pta) + (**ptb));

  system("pause");
  return 0;}

```