Design a controller for kids.   
(This is a PM interview question, so you're supposed to clarify a lot to clear ambiguity before you get to an answer)

I will start with the basic requirements and the target users.   
Target Users : Kids   
Age group: 0-3, 3- 10, 11 - 18? They are all different with different amounts of technical literacy and maturity.   
Lets consider for 0-3 and 3-10 to begin with:   
The controller should be:   
a. Light and smaller   
b. Strong - damage resistant against banging it, throwing it on the floor, etc.   
c. Water and food proof/resistant: Water and food spills shouldn't go inside it.   
d. Should be made up of non-toxic material (if possible with horrible taste but no odor so that even if kids lick it, they shouldn't be harmed)   
e. If it is wired, the juncture and the wire should be pretty strong so that it can take a good amount of stress (kids tend to swing the controllers by the wires)   
f. A small piece of material can be added so that the controller is slide proof - doesn't slide off an incline.   
g. Buttons should be easily press-able and the controller shouldn't be easy to disassemble.   
h. The buttons on the controller should be designed in accordance with the types of games that would be made for this age group.   
i. The same goes for their colors, symbols, etc.   
The I would go ahead and design something with these points in mind.

C program to Delete a node from SLL, in which the last node points to the middle node( in case of even no of nodes, it points to the first middle node) and update the SLL.

It would be a two step process.   
1. In step 1, start two pointers, one moving ahead one node at a time and the other two nodes at a time. This lets us find out the middle and the last element.   
2. In step 2, using 3-pointer mechanism that points to previous, current and next nodes, delete the given node. Handle the special case of adjusting the pointers if the deleted node is the middle node.(The middle and last-element pointers obtained in step 1 are used here) Otherwise fix, the pointers or previous, current and next nodes regularly along with fixing the pointer to the middle node from the last node.

Find the angle between the hands of a clock.

Minutes Angle = (360 \* m) / 60 = 6 \* m where m is the minutes.   
Hour Angle = ((360 \* h) / 12) + (360 \* m / 12 \* 60) = 30 \* h + m / 2   
From the above two values get the difference by subtracting the bigger value from the smaller value.   
If the difference is more than 180 then 360 - diff will be the acute angle else diff itself is the acute angle.

Calculate the number of barber's shop in your city.

This is a classic market sizing case study question. There are two approached one can take: either estimate the answer based on a small sample and extrapolate to the population of America, or estimate the supply and demand for haircuts.   
  
A sample of the first approach:   
- I live in a small town of about 10,000 people, and we have 2 barber shops in the town. Therefore, each one services roughly 5,000 people. Assuming the US population is roughly 300 million, we can estimate that there are (300,000,000 / 2,000) barber shops in the entire country, or 60,000.   
  
A sample of the second approach:   
- Assume the population is exactly half male and half female, such that there are 150 million of each gender. Splitting the population into age groups, you can estimate the number of people in each range and determine how many haircuts each group demands per year (for example, there are x people aged 0-15, x people aged 16-31, etc. and each population sample might demand a different number of haircuts/year). Once you have estimated the total demand by aggregating each population sample for both men and women, the next step is to determine how many haircuts a barbershop can supply each year. For example, it might make sense that a barber can cut mens hair at a rate of 1 per hour, and womens at a rate of 1 per 2 hours. If the average shop is open 10 hours a day, and has 2 barbers, it will give a total of 30 daily haircuts (10 male, 5 female for each of the 2 barbers). If a shop is open 5 days a week for 50 weeks, we can assume that, on average, each barbershop will supply a total of 30\*5\*50 = 7,500 haircuts per year. Using our estimated demand for haircuts/year that we calculated above, we can solve the equation that sets supply equal to demand. If, for example, our total demand was 30,000,000 haircuts per year, we could solve the following equation for "x", which equals the number of barber shops in America:   
  
7,500x = 150,000,000   
Therefore, x would be 20,000 barber shops.

algorithm - given binary tree, check if it has any 2 leaf nodes which differ by more than 1 level (write code for this)

Here's the complete code   
  
int maxDepth(Tree\* root)   
{   
if (root == NULL) return 0;   
else   
{   
int ldepth = maxDepth(root->left) + 1;   
int rDepth = maxDEpth(root->right) + 1;   
  
return ((ldepth>rDepth) ? ldepth:rdepth);   
}   
}   
  
int minDepth(Tree\* root)   
{   
if (root == NULL) return 0;   
else   
{   
int ldepth = minDepth(root->left) + 1;   
int rDepth = minDEpth(root->right) + 1;   
  
return ((ldepth>rDepth) ? rdepth:ldepth);   
}   
}   
bool isDiffMoreThanTwo(Tree\* root)   
{   
return ((maxDepth(root) - minDepth(root)) >=2 );   
}

write a code for which input is a string and set of characters acting as delimiters. Cut the given string where ever delimiters occur and return all the set of sub strings. For eg: given string abbcdeffghujsb and delimiter set:c,g,j   
then output should be: abb, deff, u, sb

Pretty much same logic as prolific coder. But in C++   
//Assumptions 2 null terminated strings. Original string will be destroyed   
//Program insert the ' ' instead of the delimiters   
  
void func(char\* str, char\* str1)   
{   
map<char,int> map1;   
int i=0;   
  
//Insert all delimiters in the map   
for(i=0;i<strlen(str1);i++)   
map1[str1[i]]++;   
  
// If this char is a delimiter, insert space instead of the char   
for(i=0;i<strlen(str);i++)   
{   
if(map1[str[i]]>0)   
str[i]=' ';   
}   
}   
  
int main(void)   
{   
char str[]= "abhishekagrawal";   
char str1[]="askw";   
printf("Before %s %s\n",str,str1);   
func(str,str1);   
printf("After %s \n",str);   
  
cin.get();   
return 0;   
}

puzzle - 2 trains traveling in opposite direction, bird starts from one and reaches other and flies back and forth like this till train collide. Find the time taken to collide, total distance traveled by the bird, and no. of times bird makes a U turn.

D: distance between 2 trains   
v1,v2: speeds of two trains   
v3: speed of bird(v3>v1 &&v3>v2)   
  
1)Find the time taken to collide   
T1=D/(v1+v2)   
  
2)total distance traveled by the bird   
D2=T1\*v3   
  
3)# of times bird makes a U turn   
Maybe infinite (not sure)

Design an alarm clock for the blind. They asked how the person could identify the physical object.

Questions:   
-Is this person partially or fully blind? Could use bright lights to guide partially blind   
-Is the customer accustomed to using an alarm regular alarm clock? If they're familiar with a regular alarm clock, we can improve on the existing design and maintain familiarity   
-What is the budget? This would be important to see if we have to build a nuts and bolts alarm clock or one powered by software that could respond to voice/sound commands or even a motion detector   
-Where would the alarm clock be located? Next to the bed (most likely but don't assume) or across the room?   
-How will it be powered? Manual key or electric   
  
Other factors:   
-It's easy for a blind person to knock off an upright alarm clock, so we may want to design something with a flatter design   
-Regardless of design, we would use Braille lettering   
-Simple actions for alarm on-off or easy to locate buttons   
  
Alternatives:   
-If it responds to voice/sound or motion detector that opens up other design possibilities